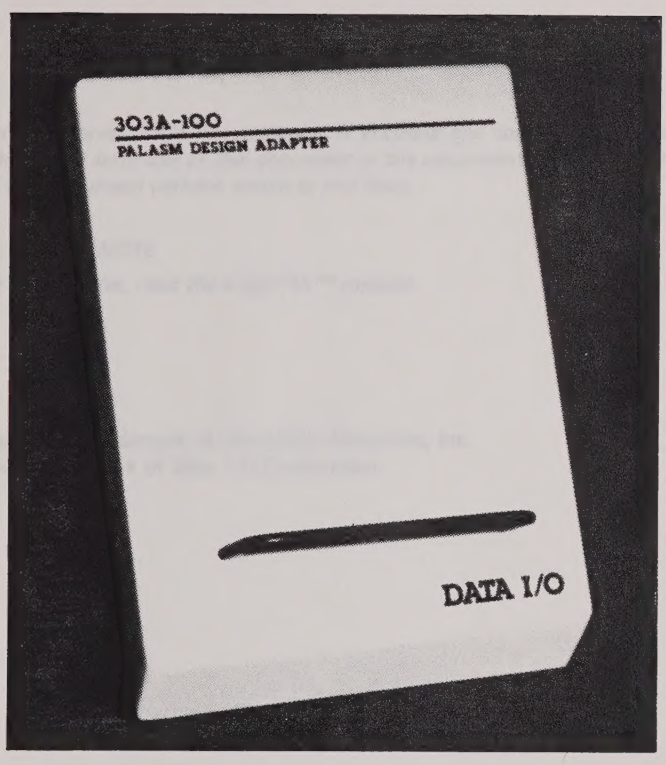


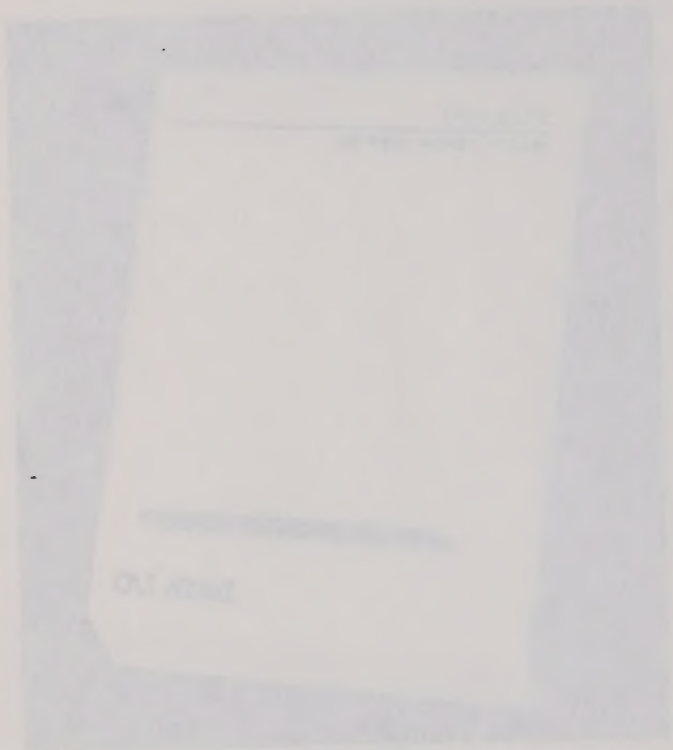
303A-100 PALASM DESIGN ADAPTER 715-1114-001

REV A AUG 83

10-715-1114

(303A-100)

Copyright © Data I/O Corporation, 1983. All rights reserved.



Data I/O has made every attempt to ensure that the information in this document is accurate and complete. However, Data I/O assumes no liability for errors, or for any damages that result from use of this document or the equipment which it accompanies. Data I/O reserves the right to make changes to this document without notice at any time.

NOTE

Before using this adapter, read the LogicPak™ manual.

PAL® is a registered trademark of Monolithic Memories, Inc.
LogicPak™ is a trademark of Data I/O Corporation.

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_11482

TABLE OF CONTENTS

SECTION 1. INTRODUCTION

1.1 OVERVIEW 1-1

1.2 APPLICATIONS 1-1

1.3 OPERATIONAL OVERVIEW 1-2
1.3.1 PAL Assembler (PALASM) 1-6

1.4 SPECIFICATIONS 1-11

1.5 FIELD APPLICATIONS SUPPORT 1-11

1.6 WARRANTY 1-11

1.7 SERVICE 1-11

1.8 ORDERING 1-11

SECTION 2. INSTALLATION

2.1 INSPECTION 2-1

2.2 ADAPTER INSTALLATION 2-1

2.3 ADAPTER REMOVAL 2-2

2.4 SERIAL INTERFACING 2-3

2.5 REPACKAGING FOR SHIPMENT 2-5

SECTION 3. OPERATION

3.1 OVERVIEW 3-1
3.1.1 Data Entry 3-4

3.2 POWER UP 3-5

3.3 POWER DOWN 3-5

3.4 PALASM-SPECIFIC COMMANDS 3-6
3.4.1 Family Code and Pinout Code Selection 3-6
3.4.2 Receive PALASM Source 3-7
3.4.3 Transmit PALASM Source 3-10
3.4.4 Assemble PALASM Source 3-11
3.4.5 Simulate Function Table 3-14
3.4.6 Edit Source 3-17

3.5 SYSTEM COMMANDS 3-25
3.5.1 System Commands 3-26
3.5.2 Display Command Menu 3-26
3.5.3 Family Code and Pinout Code 3-27
3.5.4 Set Reject Count Option 3-28

3.5.5 Select Verify Option 3-29
3.5.6 Select Security Fuse Option 3-30
3.5.7 Enter Functional Test Data 3-32
3.5.8 Display Fuse Pattern 3-37
3.5.9 JEDEC Format Data Exchange 3-40
3.5.10 Edit Fuse Pattern 3-47
3.5.11 Display Configuration Number 3-50
3.5.12 Select Attributes 3-51
3.5.13 Exit Commands 3-53

SECTION 4. CIRCUIT DESCRIPTION

4.1 INTRODUCTION 4-1

4.2 GENERAL ARCHITECTURE 4-1

4.3 COMPONENT LAYOUT 4-1

4.4 FIRMWARE OPERATION 4-2

APPENDIX A. STANDARD DATA TRANSFER FORMAT BETWEEN DATA PREPARATION SYSTEM AND PROGRAMMABLE LOGIC DEVICE PROGRAMMER

A.1 INTRODUCTION A-2
A.1.1 BNF Rules A-2
A.1.2 BNF Definitions A-2

A.2 TRANSMISSION PROTOCOL A-3
A.2.1 Design Spec A-3
A.2.2 Transmission Check-Sum A-3
A.2.3 Fields A-3

A.3 FORMAT FIELD DEFINITIONS A-5
A.3.1 Device to Be Programmed A-5
A.3.2 Fuse Information A-5
A.3.3 Structured Functional Test Information A-5
A.3.4 Optional Information A-6

A.4 OTHER RULES A-7
A.4.1 Transportability A-7
A.4.2 Restrictions and Variations A-7

APPENDIX B. DEVICE INFORMATION/ERROR CODES B-1

LIST OF FIGURES

1-1 303A-100 PALASM Design Adapter 1-1
1-2 PALASM Operational Overview 1-2
1-3 Main Command Menu 1-2
1-4 Basic Gates Example 1-3
1-5 Logic Diagram for Basic Gates Example 1-4
1-6 PALASM Source File for Basic Gates Example 1-5
1-7 Fuse Pattern 1-5
1-8 Test Vectors From Basic Gates Example 1-5
1-9 Source File Heading 1-6
1-10 Examples of Pin Names 1-6
1-11 Sum-of-Products Expressions 1-7
1-12 Example of Expression for Exclusive-OR-Type PAL 1-8
1-13 Fixed Symbols 1-8
1-14 Simulation Example 1-10
2-1 Adapter Installation 2-1
2-2 Sample Interconnection Methods 2-3
3-1 PALASM Operational Overview 3-1
3-2 Programmer Power Switch Location 3-5
3-3 Receive Source File 3-7
3-4 Assembling a Source File With Errors 3-12
3-5 Main Command Menu 3-26
3-6 Prompts for Entering Functional Test Data 3-34
3-7 Entering Functional Test Data 3-34
3-8 Complete Fuse Pattern 3-37
3-9 Logic Diagram for Basic Gates Example 3-38
3-10 JEDEC Transmission—Basic Gates Example 3-40
3-11 JEDEC Format (Breakdown of Figure 3-10) 3-41
3-12 Computing the Transmission Checksum 3-42
3-13 Computing the Fuse RAM Checksum 3-43
3-14 Default Fuse Editor Pattern 3-48
3-15 Starting Fuse Not on Row Boundary 3-49
4-1 Block Diagram, PALASM Design Adapter 4-1
4-2 Test Vector Memory Map 4-2
4-3 Test Source Buffer 4-2
B-1 Logic Diagram PAL10H8 ..... B-6
B-2 Logic Diagram PAL10L8 ..... B-7
B-3 Logic Diagram PAL12H6 ..... B-8
B-4 Logic Diagram PAL12L6 ..... B-9
B-5 Logic Diagram PAL14H4 ..... B-10
B-6 Logic Diagram PAL14L4 ..... B-11
B-7 Logic Diagram PAL16A4 ..... B-12
B-8 Logic Diagram PAL16C1 ..... B-13
B-9 Logic Diagram PAL16H2 ..... B-14
B-10 Logic Diagram PAL16H8 ..... B-15
B-11 Logic Diagram PAL16HD8 ..... B-16
B-12 Logic Diagram PAL16L2 ..... B-17
B-13 Logic Diagram PAL16L8 ..... B-18
B-14 Logic Diagram PAL16LD8 ..... B-19
B-15 Logic Diagram PAL16P8 ..... B-20
B-16 Logic Diagram PAL16R4 ..... B-21
B-17 Logic Diagram PAL16R6 ..... B-22
B-18 Logic Diagram PAL16R8 ..... B-23
B-19 Logic Diagram PAL16X4 ..... B-24
B-20 Logic Diagram PAL12H10 ..... B-25
B-21 Logic Diagram PAL12L10 ..... B-26
B-22 Logic Diagram PAL14H8 ..... B-27
B-23 Logic Diagram PAL14L8 ..... B-28

B-24 Logic Diagram PAL16H6 ..... B-29
B-25 Logic Diagram PAL16L6 ..... B-30
B-26 Logic Diagram PAL18H4 ..... B-31
B-27 Logic Diagram PAL18L4 ..... B-32
B-28 Logic Diagram PAL20H2 ..... B-33
B-29 Logic Diagram PAL20L2 ..... B-34
B-30 Logic Diagram PAL20L8 ..... B-35
B-31 Logic Diagram PAL20L10 ..... B-36
B-32 Logic Diagram PAL20R4 ..... B-37
B-33 Logic Diagram PAL20R6 ..... B-38
B-34 Logic Diagram PAL20R8 ..... B-39
B-35 Logic Diagram PAL20X4 ..... B-40
B-36 Logic Diagram PAL20X8 ..... B-41
B-37 Logic Diagram PAL20X10 ..... B-42

LIST OF TABLES

1-1 Using the PALASM Design Adapter Manual 1-1
1-2 List of Valid Test Conditions 1-9
2-1 Null Count/Send Chart 2-4
3-1 PLDS System Command Summary 3-2
3-2 Source Buffer Size 3-8
3-3 PALASM Assembler Error Messages 3-13
3-4 Simulator Error Messages 3-16
3-5 Editor Commands 3-18
3-6 Vector Editor Command Characters 3-35
3-7 Vector Symbol Definition 3-36
3-8 Translation Input Errors 3-44
3-9 ASCII Values of Field Identifiers 3-45
3-10 Translator Input Error Codes 3-45
4-1 PALASM Design Adapter Memory Map 4-2
4-2 Fuse RAM Memory Map 4-2
4-3 Test Information Memory Map 4-2
B-1 PALASM Family and Pinout Codes B-3
B-2 PLDS Error Codes B-4

SECTION 1
INTRODUCTION

1.1 OVERVIEW

The 303A-100 PALASM (PAL Assembler and Simulator) Design Adapter, shown in figure 1-1, is a complete design tool that enables you to develop logic equations and function tables for programming and testing Programmable Array Logic (PAL®) devices. The design adapter includes a text editor for creating a PALASM source file and the means of transferring this text between the Programmable Logic Development System (PLDS) and a host computer. The PALASM program will convert a source file into the fuse pattern and structured test vectors required by the programming/testing adapters. The fuse and testing data may be transferred to a host computer or other programmers in the JEDEC ASCII-logic format. A fuse editor and a functional test data editor are also integral parts of the PALASM design adapter.

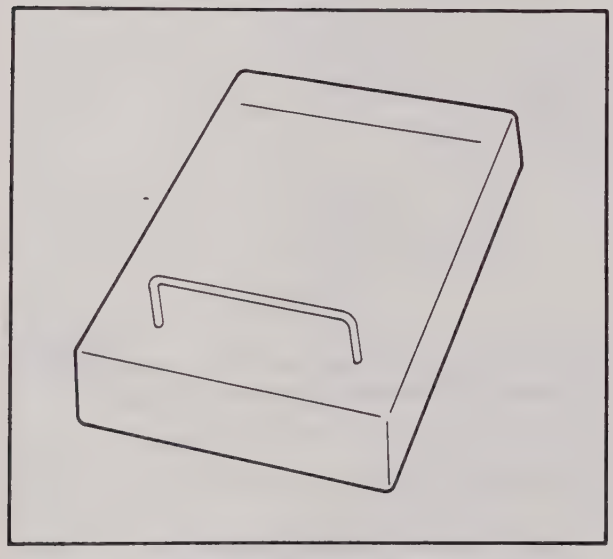


Figure 1-1. 303A-100 PALASM Design Adapter

This manual describes how to use the PALASM Design Adapter. Subjects addressed in this manual and their corresponding sections are listed in table 1-1. Use this table as a quick reference point for the major sections. In this manual, we will refer to the operational procedures for the 29A Universal Programmer. See your programmer manual for System 19 and 100A key sequences. The entries that you are to make from either the programmer or the terminal are indicated by the entry enclosed in a key symbol. For example,

Table 1-1. Using the PALASM Design Adapter Manual

SUBJECT	SECTION
Applications	1.2
Installation procedures for Design Adapter	2.2
Basic operation instructions	3.0
System commands	3.5
Error codes	Appendix B
Circuit description	4.0
Family and pinout codes	Appendix B
Logic diagrams	Appendix B
Data I/O Service Centers	Back of manual
Warranty information	Back of manual

indicates that the ESCAPE key on the terminal keyboard should be pressed. Also, the symbols shown below indicate modes of operation.

Terminal Mode **Programmer Front Panel Mode**

1.2 APPLICATIONS

The PALASM design adapter supports PALs supplied by several semiconductor manufacturers. Application notes and design information are available from these manufacturers and their distributors. Your Data I/O sales representative can provide additional information or you may refer to one of the following handbooks:

MMI PAL HANDBOOK, available from Monolithic Memories, Inc., 1165 Arques Avenue, Sunnyvale, CA 94086

AMD PAL HANDBOOK, available from Advanced Micro Devices, 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088

The family code and pinout code table (table B-1 in appendix B) lists all the devices that are supported by this adapter. This table will be updated as new devices are added. As Data I/O increases the capabilities of the LogicPak™ to support new devices, firmware and/or hardware updates will be available for existing adapters to add new devices to existing device families. Contact the Data I/O Customer Service department for the latest revision and any required firmware updates.

1.3 OPERATIONAL OVERVIEW

Data development with the 303A-100 PALASM Design Adapter is accomplished with a PLDS configuration consisting of the LogicPak™, design adapter, terminal, and a Data I/O programmer. The terminal, though not essential for many operations, is required for entering design information and other design adapter operations. (See section 3 for terminal operation.)

Figure 1-2 shows the functions unique to the PALASM system in a typical operational sequence. The PALASM functions common to the LogicPak™ (such as receiving and transmitting JEDEC-format data, vector and fuse editing, and selecting test options) are not shown, but may be utilized in the same way as the LogicPak™. Figure 1-3 shows the menu of all PALASM system functions, and sections 3.4 and 3.5 explain the functions in detail.

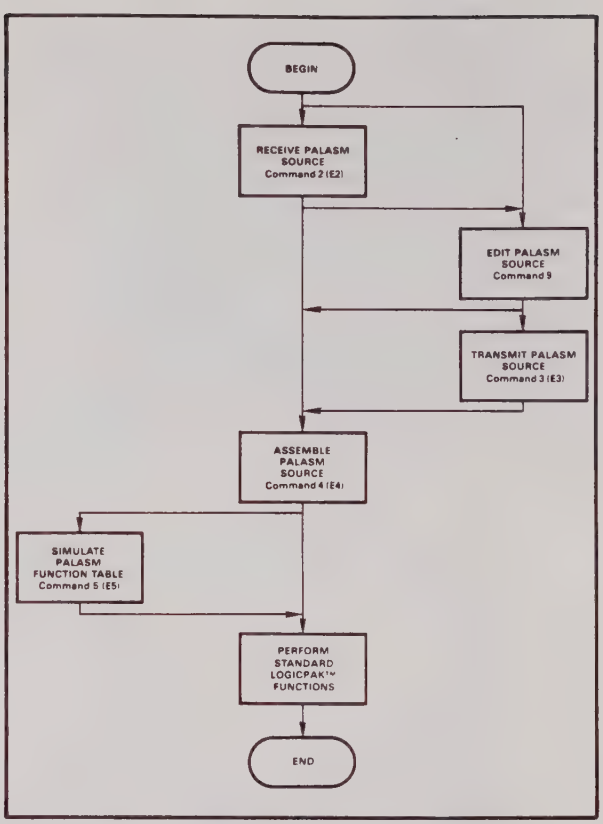

BEGIN
RECEIVE PALASM SOURCE Command 2 (E2)
EDIT PALASM SOURCE Command 9
TRANSMIT PALASM SOURCE Command 3 (E3)
ASSEMBLE PALASM SOURCE Command 4 (E4)
SIMULATE PALASM FUNCTION TABLE Command 5 (E5)
PERFORM STANDARD LOGICPAK™ FUNCTIONS
END

Figure 1-2. PALASM Operational Overview

```
DATA I/O CORP. - PALASM Design Adapter - 303A-100-V02  (C) 1982,1983

      - GENERAL COMMANDS -                    - I/O COMMANDS -
0 - Display menu                        2 - Receive PALASM source
1 - Enter family/pinout code            3 - Transmit PALASM source
6 - Enter verify option                 B - Receive JEDEC data
7 - Enter security fuse option          C - Transmit JEDEC data
8 - Enter functional test data
F - Configuration number
G - Select attributes

   - SOURCE EQUATION COMMANDS -             - FUSE MAP COMMANDS -
4 - Assemble PALASM source              A - Display fuse pattern
5 - Simulate function table             D - Display fuse sumcheck
9 - Edit source                         E - Edit fuse pattern

NOTE - Always transmit an "ESC" before removing adapter
```

Figure 1-3. Main Command Menu

As can be seen from figure 1-2, data development typically begins by using a PALASM source file to describe the function you want the device to perform. Figure 1-4 shows example Boolean equations (source equations) for a PAL12H6, along with conventional logic symbology, the PAL Logic symbology, and the pinout for a PAL12H6. This figure demonstrates how programmable logic can implement the basic inverter, AND, OR, NAND, NOR, and exclusive-OR functions.

To program the PAL12H6, the Boolean equations must be translated into a fuse pattern. This fuse pattern may be generated using PALASM. The binary fuse pattern signifies which fuses need to be blown in order for the PAL12H6 to perform the functions specified by the equations.

Traditionally, an engineer performed this translation step by following the lines on the logic diagram for the PAL12H6 (see figure 1-5) to the intersection where the engineer wanted the fuse to remain intact, and marked it with an "X" (the blank diagram shows all fuses blown). Thereby, it was possible to develop a fuse pattern that would be programmed into the PAL12H6. The pattern would then have to be manually translated to a memory image suitable for programming the device.

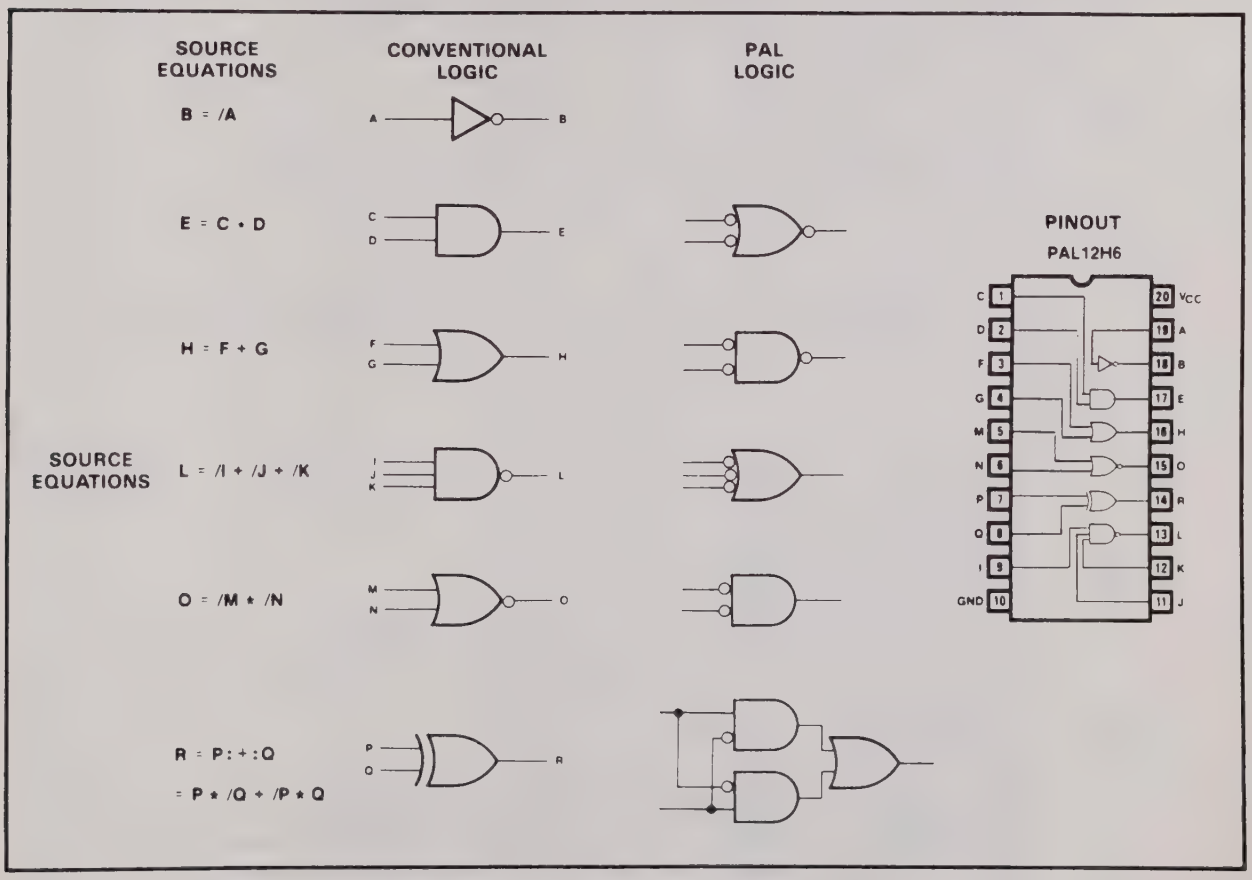


Figure 1-4. Basic Gates Example

1.3.1 PAL ASSEMBLER (PALASM)

Figure 1-6 is a sample PALASM source file (design specification). This example illustrates the use of programmable logic to implement the basic gates: inverter, AND gate, OR gate, NAND gate, NOR gate, and exclusive-OR gate. The source equations define the required logic functions, and the function table exercises all inputs and tests basic function performance.

The PALASM input file is composed of five main sections: the heading, the pin list, the source equations, the function table, and the description. These are discussed in the subsections which follow.

Heading

The first four lines of the source file make up the heading. Line one must have a valid PAL device name, starting in column one. The rest of the heading is optional, but a recommended format is shown in figure 1-9.

NOTE

The device number (XXXXX in figure 1-9) must start in column 4. See table B-1 in appendix B for a list of devices supported.

Pin List

The pin list starts on line five. It consists of a name for each pin on the device (20 or 24 names). The names must be in pin number sequence starting from pin 1. The list may continue on additional lines if required. To ensure compatibility with all versions of PALASM, the following points should be remembered:

- The names should not exceed eight characters.
- The names should use only the letters A to Z, the digits 0 to 9, the underscore (_), and the period (.).
- A leading slash (/) should be used to indicate an active low signal.
- Characters other than those listed above are either reserved or not recommended.
- Duplicate pin names are allowed, but should not be used in any equations. A common use of this feature is for "no connects" (NC).
- The names "Q0" and "/Q0" refer to the same pin, Q0 indicating active high, and /Q0 indicating active low.

Figure 1-10 lists some pin name characteristics and examples.

```
PALxxxxx                          PAL DESIGN SPECIFICATION
USER'S PART NUMBER                DESIGNER'S NAME     DATE
DEVICE APPLICATION NAME
COMPANY NAME   ADDRESS
```

Figure 1-9. Source File Heading

Recommended valid characters:	A...Z 0...9 _ .
Reserved characters:	* + = - : / ( )
Valid names:	CLK ADDR12 DATA_ENA Q0 /Q0
	_8251CS NC VCC GND .RESET
Invalid names:	DATA(1) -GATE BASE+23 DATAENA- R/W
Not recommended:	74$00 &GATE AVERYLONGNAME

Figure 1-10. Examples of Pin Names

Source Equations

The general form of the equation is as follows:

```
IF( PRODUCT) OUTPUT_NAME = EXPRESSION ; COMMENT
```

Programmable I/O. The "IF(PRODUCT)" is used to program the three-state output control on programmable I/O PALs (i.e., PAL16L8, PAL16R4). If the "IF(PRODUCT)" is omitted, the output will be always enabled.

```
      IF(VCC) DATA1 = A * B ; OUTPUT ONLY
              DATA1 = A * B ; OUTPUT ONLY
      IF(GND) DATA1 = A     ; INPUT ONLY WITH DUMMY INPUT
IF(CS * GATE) DATA1 = A * B ; INPUT / OUTPUT
```

Active High and Low Outputs. If an output is to be active low (either programmable or fixed), there must be a "phase inversion" between the pin list name and the name as it appears in its defining source equation. Phase inversion means to place a leading slash (/) before only one of either the pin list name or the name in the source equation. For example, PAL12H6 is an active high device; therefore, the pin list names and the equation names for all outputs must have the same phase. If the device is active low (such as PAL12L6), there must be a phase difference. To illustrate, if the pin list name for an output pin is "DATA1", the defining source equations would appear as follows for the PAL12H6 and PAL12L6:

Pin list ....DATA1....

Equation for PAL 12L6 (an active low device)	/DATA1 = A * B (Note phase inversion.)
Equation for PAL12H6 (an active high device)	DATA1 = A * B

Registered Outputs. Registered outputs use a ":=" instead of the "=". Therefore, "/DATA2 := A * B" denotes a registered output, while "/DATA1 = A * B" is not registered.

Sum-of-Products Expressions. The right side of the PALASM source equation must be a sum-of-products expression. The asterisk (*) (as shown in examples 1 and 2 in figure 1-11) is used for the AND operator, and the plus (+) is used for the OR operator. The semicolon (;) is used to indicate that the rest of the line is a comment.

PALASM will not expand terms enclosed within parentheses. The only case where parentheses are allowed on the right side of the equation is in the fixed symbols of the PAL16A4 and PAL16X4, as described below. (See figure 1-11, example 3, for examples of correct and incorrect equations.)

Example 1:

```
/CS1  =  A15 *  A14 *  A13 * /A12  ; E000 HEX
      +  A15 *  A14 * /A13 *  A12  ; D000 HEX
      +  A15 * /A14 *  A13 * /A12  ; A000 HEX
```

Example 2:

```
INHIBIT = SENSE1 + SENSE2 + SENSE3 * ENAB2
```

Example 3:

```
/CS  = A15 * (A14 + /A13)       ; INCORRECT

/CS  = A15 * A14 + A15 * /A13   ; CORRECT
```

Figure 1-11. Sum-of-Products Expressions

Exclusive-OR. The exclusive-OR symbol (:+:) may be used only in equations for outputs that include an exclusive-OR gate (i.e., PAL16X4, PAL20X8, etc.). PALASM will not expand exclusive-OR equations into the required AND/OR equations for non-exclusive-OR PALs. (See figure 1-12 for an example.)

```
/D0 :=  /I1 * /I0 * /D0          :HOLD D0
     +  /I1 *  I0 * /D1          :SHIFT RIGHT
    :+:  I1 * /I0 * /LIRO        :SHIFT LEFT
     +   I1 *  I0 * /D0          :LOAD D0
```

Figure 1-12. Example of Expression for Exclusive-OR-Type PAL

Fixed Symbols (Arithmetic-Gated Feedback). The fixed symbol is a special feature of the PAL16A4 and PAL16X4. These devices have additional logic that combines input and output pairs to produce internal "inputs," which may be referred to by means of the fixed symbols. A fixed symbol itself is a parenthesized expression describing the desired arithmetic combination of the selected input and output pair. Within a fixed symbol, the registered outputs on pins 17, 16, 15, and 14 must be named A0, A1, A2, and A3 and the inputs on pins 4, 5, 6, and 7 must be named B0, B1, B2, and B3. All valid combinations are shown in figure 1-13. The fixed symbols deal with fixed input and output pairs. Therefore, mixing numbers such as in "(A3*B2)" is not allowed. The fixed symbols may be used in equations as any other input signal. (See figure 1-13 for examples.)

Function Table

The function table section of the PALASM source file (such as the example in figure 1-6) is used to simulate the operation of the PAL design and produce structured test vectors that may be used with the P/T adapters. The PLDS simulation compares the inputs and outputs expressed in the function table to the computed values based on the fuse pattern in the fuse RAM. (See figure 1-7 for an example of the fuse pattern that corresponds to the source equations shown in figure 1-6.) The fuse pattern could result from assembling source equations or from any other method of loading the fuse RAM. With a fuse pattern and a function table loaded into the PLDS, a simulation may be performed. Figure 1-8 shows the test vectors resulting from a simulation of the function table in figure 1-6 and the fuse pattern of figure 1-7. The FORTRAN versions of PALASM operate on the source equations directly, not on a fuse pattern.

The function table is a valuable design tool because it detects common errors (signal inversions and typing errors) and also provides a check on more subtle logic errors.

The beginning of the function table is defined by the key words, "FUNCTION TABLE," in column one (see line 0019 in figure 1-6). A list of pins to be tested follows and must be terminated by a dashed line (starting in column 1). The names may have a different phase (i.e., "D1" and "/D1") and may be in a different sequence from the equation pin list starting on line 5. If a phase inversion exists between a pin name in the equation pin list and the same name in the function table pin list, then all states for that pin in the function table must be inverted from the logic in the source equations.

Valid Combinations of Names for Fixed Symbols (where n is 0, 1, 2, 3)

(An)	(/An)	(Bn)	(/Bn)
(An+Bn)	(/An+Bn)	(An+/Bn)	(/An+/Bn)
(An*Bn)	(/An*Bn)	(An*/Bn)	(/An*/Bn)
(An:+:Bn)	(An:*:Bn)		

Fixed Symbol Expressions

```
/A3 := /I2 * /I1 * (A0) + /I2 * I1 * (B1)

 L  = (A3*/B3) + (A3:*:B3) * (/A2*B2)
```

Figure 1-13. Fixed Symbols

Each test line (vector) must have a test condition for each pin named in the function table pin list (see table 1-2 for a list of valid test conditions). Comments may be placed after the test conditions (the semicolon is recommended for the delimiter). Blank lines or lines with just a comment may also be placed in the table.

Table 1-2. Valid Test Conditions

TEST CONDITIONS	INPUT	OUTPUT
L logic level zero	drive low	test for zero
H logic level one	drive high	test for one
X don't care	not defined	don't test
Z high impedence	not defined	test for high Z
C clock	clock	N/A

NOTE

The FORTRAN versions of PALASM assume "X" to be a logic level zero input. This is not a valid assumption for the PLDS simulator or the P/T adapters. Remember, "don't care" means you don't care what level is applied to an input. If a logic level zero is desired, use an "L".

Figure 1-14 shows the source file, assembly, and simulation for a PAL with both registered and combinatorial outputs. The first line in the function table generates vector 0001, the second line, vector 0002, and so on. The two types of output pin (combinatorial and registered) are specified differently in the function table, as shown below:

- **Combinatorial output**—Combinatorial outputs may depend on input pin states or fed-back registered outputs or both. The combinatorial output tested on a given function table line is the result of input pin states and fed-back registered output states from the same line. The registered output states on the same line are defined as "next states." See outputs E1 and E2 in figure 1-14.
- **Registered output**—Registered outputs may also depend on input pin states, fed-back registered outputs, or both. However, testing a registered output on a given line of the function table involves applying the input pin states from the same line and any fed-back registered outputs from the previous line. Registered output states from the previous line are defined as "present states." See outputs Q1 and Q2 in figure 1-14. Note that no present states are defined for the first line of the function table, so registered outputs that require present states must appear as "X" on the first line (e.g., Q2 in figure 1-14).

Description

This section starts with the key word, "DESCRIPTION," in column one, as shown in figure 1-6. You should describe the operation and application of the PAL design in this section. This enables the PALASM source file to completely document the custom design.

```
 Command :  3 - Transmit PALASM source
PAL16R4                                        PAL DESIGN SPECIFICATION
N/A                                            M HOLLEY        31 MAY 83
EXAMPLE FOR FUNCTIONAL TEST TABLE
DATA I/O
CLK  D1 D2 NC NC NC NC NC NC GND /OE E1 E2 Q1 Q2 NC NC E3 NC VCC

         /Q1 := D1

         /Q2 := /Q1

         /E1  = D1

         /E2  = /Q1

IF(D2)   /E3  = /Q2

FUNCTION TABLE

/OE CLK D2  D1   E1   E2   E3   Q1   Q2
-------------------------------------------------------------
 L   C   H   L    H    H    X    H    X
 L   C   H   L    H    H    H    H    H
 L   C   H   H    L    L    H    L    H
 L   C   H   H    L    L    L    L    L
 L   L   H   L    H    L    L    L    L   ; NO CLOCK
 L   C   H   L    H    H    L    H    L
; HIGH IMPEDANCE TEST
 H   L   H   L    H    X    X    Z    Z
 L   L   L   L    H    X    Z    H    X
-------------------------------------------------------------

DESCRIPTION

AN EXAMPLE TO SHOW THE SIMULATOR OPERATION WITH REGISTERED PAL'S.
```

```
 Command :  4 - Assemble PALASM source
0250 Fuses to be blown
```

```
 Command :  5 - Simulate function table
0001 C01XXXXXXN0HHHXXXXXN
0002 C01XXXXXXN0HHHHXXHXN
0003 C11XXXXXXN0LLLHXXHXN
0004 C11XXXXXXN0LLLLXXLXN
0005 001XXXXXXN0HLLLXXLXN
0006 C01XXXXXXN0HHHLXXLXN
0007 001XXXXXXN1HXZZXXXXN
0008 000XXXXXXN0HXHXXXZXN
```

Figure 1-14. Simulation Example

1.4 SPECIFICATIONS

The adapter receives its power through the LogicPak™ and in the adapter. The physical and environmental specifications of the adapter are:

- altitude (operating): sea level to 3 km (10,000 ft)
- humidity (operating): 90% maximum (noncondensing)
- humidity (storage): 95% maximum (noncondensing)
- temperature (operating): 5° to 45°C (41° to 113°F)
- temperature (storage): −40° to 70°C (−40° to 158°F)
- weight: 0.255 kg (9 oz.)
- dimensions: 16.6 cm x 12.3 cm x 2.1 cm (6.54 in. x 4.84 in. x 0.81 in.)

1.5 FIELD APPLICATIONS SUPPORT

Data I/O has field applications engineers throughout the world. They can provide additional information about interfacing Data I/O products with other systems and answer questions about your equipment.

These engineers are located within the United States at the addresses listed in the back of this manual. For international applications support, contact your nearest Data I/O representative.

1.6 WARRANTY

The 303A-100 adapter is warranted against defects in materials and workmanship. The warranty period of 90 days begins when you receive the equipment. The warranty card inside the back cover of this manual explains the length and conditions of the warranty. For warranty service, contact your nearest Data I/O Service Center.

1.7 SERVICE

Data I/O maintains service centers throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. A list of all service centers is located in the back of this manual.

1.8 ORDERING

To place an order for equipment, contact your Data I/O sales representative. Orders for shipment must include:

- A description of the equipment (see the latest Data I/O price list or contact your sales representative for equipment and part numbers)
- purchase order number
- desired method of shipment
- quantity of each item ordered
- shipping and billing address of the firm, including ZIP code
- name of person ordering the equipment

SECTION 2

INSTALLATION

2.1 INSPECTION

The adapter was thoroughly tested and inspected before shipment and was carefully packaged to prevent shipping damage. Inspect your adapter to ensure that no damage occurred during shipment. If you notice any damage, file a claim with the carrier and notify Data I/O.

2.2 ADAPTER INSTALLATION

To insert the adapter into the LogicPak™:

1. Align the guide pins on the underside of the adapter with the guide pin holes on the LogicPak™ (see figure 2-1).
2. Gently set the adapter on the LogicPak™.
3. Firmly press down on the front edge of the adapter to lock the connector pins into the connector receptacle (see figure 2-1).

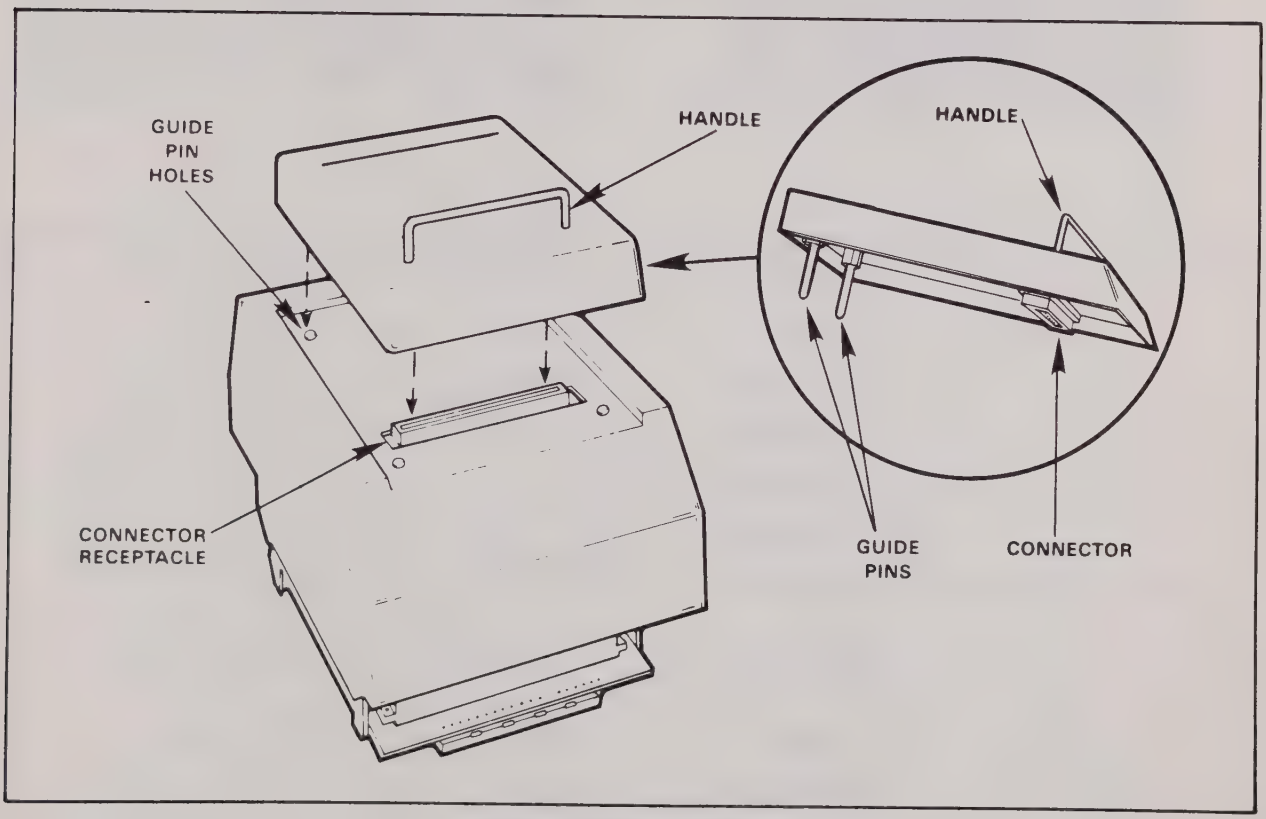


Figure 2-1. Adapter Installation

2.3 ADAPTER REMOVAL

CAUTION

BEFORE REMOVING THE ADAPTER, you MUST return control to the programmer.

This is done by pressing the ESC key at the terminal, the KEYBOARD key on the System 19, or the VERIFY key on the 100A or 29A.

Because the processor in the programmer executes firmware resident in the adapter, this precaution must be taken before removing the adapter from the LogicPak™ to prevent program interruption and possible loss of RAM data.

To remove the adapter:

1. Ensure that the PLDS has completed the current operation.
2. Return control to the programmer as described above. Ensure that the programmer display indicates programmer control.
3. While holding down the LogicPak™, grasp the adapter handle and gently remove the adapter.

2.4 SERIAL INTERFACING

The LogicPak™ uses the RS-232C serial interface of the programmer for interaction with a terminal and for communication with a host computer. Figure 2-2 illustrates some typical interconnection methods.

The five-wire handshake interconnection may be used to upload (data flow from LogicPak™ to host) or download (data flow from host to LogicPak™) data at any supported baud rate with a host system that recognizes the five-wire protocol (see figure 2-2a).

Software handshake (CTRL S, DC3, or ASCII hex 13 to stop transmission and CTRL Q, DC1, or ASCII hex 11 to resume) may be used with either the five-wire or the three-wire interconnection (figure 2-2a or b), but will be recognized by the LogicPak™ only while uploading. CTRL Y (EM, or ASCII hex 19) may be sent by the host to terminate an upload.

NOTE

Due to processing overhead, I/O overrun errors may occur when downloading JEDEC files (see section 3.5.9) over a three-wire link at baud rates above 4800.

The control characters described above have a similar effect when entered from a terminal: CTRL S AND CTRL Q will halt and resume output to the display, and CTRL Y will terminate a command during display output.

NOTE

An ESC character (ASCII hex 1B) received at any time will terminate all LogicPak™ operations immediately and return control to the programmer.

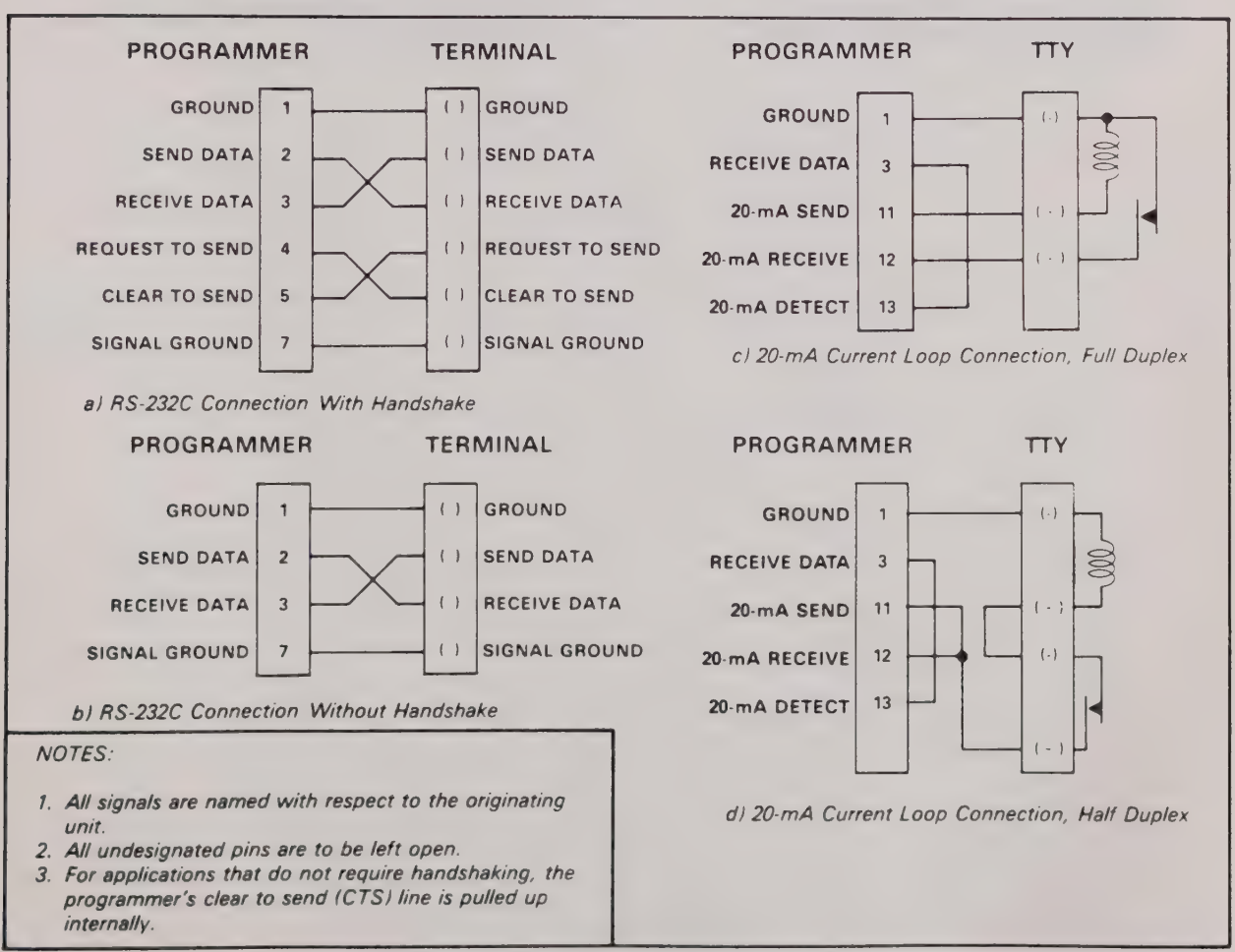

PROGRAMMER
TERMINAL
GROUND 1 GROUND
SEND DATA 2 SEND DATA
RECEIVE DATA 3 RECEIVE DATA
REQUEST TO SEND 4 REQUEST TO SEND
CLEAR TO SEND 5 CLEAR TO SEND
SIGNAL GROUND 7 SIGNAL GROUND
a) RS-232C Connection With Handshake
PROGRAMMER
TERMINAL
GROUND 1 GROUND
SEND DATA 2 SEND DATA
RECEIVE DATA 3 RECEIVE DATA
SIGNAL GROUND 7 SIGNAL GROUND
b) RS-232C Connection Without Handshake
PROGRAMMER
TTY
GROUND 1
RECEIVE DATA 3
20-mA SEND 11
20-mA RECEIVE 12
20-mA DETECT 13
c) 20-mA Current Loop Connection, Full Duplex
PROGRAMMER
TTY
GROUND 1
RECEIVE DATA 3
20-mA SEND 11
20-mA RECEIVE 12
20-mA DETECT 13
d) 20-mA Current Loop Connection, Half Duplex
NOTES:
1. All signals are named with respect to the originating unit.
2. All undesignated pins are to be left open.
3. For applications that do not require handshaking, the programmer's clear to send (CTS) line is pulled up internally.

Figure 2-2. Sample Interconnection Methods

The LogicPak™ will transmit the line feed character (LF, or ASCII hex 0A) and a variable number of NULL characters (ASCII hex 00) following every RETURN character (ASCII hex OD), based on the null count entered using select function D9 on the System 19 or 29A (see table 2-1).

Table 2-1. Null Count/Send Chart

Null Count[a]	LF Sent?	Number of Nulls Sent
00-7F	Yes	Same as null count
80-FE	No	Null count minus 80 hex
FF	No	None

[a] See your programmer manual for details on entering the null count.

NOTE

The NULL count option is not available when the LogicPak™ is installed in a Model 100A. A line feed and four NULLs are always sent.

To set the baud rate, parity, and stop bits, refer to your programmer manual; we recommend a transmission rate of 9600 baud.

NOTE

Be sure the programmer power is off before setting the parity and stop bits.

To set up your terminal:

1. Disable any special terminal functions that use CTRL Z, CTRL C, CTRL Y or CTRL P.
2. Select the cursor mode that deletes the character in the current cursor position (i.e. destructive cursor). This will vary from terminal to terminal.
3. The default line mode of the LogicPak™ varies with its system as follows:
   - Full duplex: System 19.
   - Half duplex: System 29A and 100A.

   The line mode of the LogicPak™ may be changed at any time (see section 3.5.12).

2.5 REPACKAGING FOR SHIPMENT

If the adapter is to be shipped to Data I/O for service or repair, attach a tag to it describing the work required and identifying the owner. In correspondence, identify the unit by part number, revision level, and name of the unit. If the original shipping container is to be used, place the adapter in the container with the appropriate packing materials, and seal the container with strong tape. If another container is used, be sure that it is a heavy carton, wrapped with heavy paper or plastic; use appropriate packing material, and seal well with strong tape. Mark the container "DELICATE INSTRUMENT" or "FRAGILE."

SECTION 3

OPERATION

3.1 OVERVIEW

This section explains operation of the PLDS with the PALASM design adapter installed. Sections 3.2 and 3.3 cover the basic operations to apply and remove power to the system. Section 3.4 explains the commands specific to PALASM. Section 3.5 covers the system commands that PALASM shares in common with the LogicPak™. See figure 3-1 for a flowchart of typical operation, and table 3-1 for a summary of all system commands and the sub-commands available within certain operations.

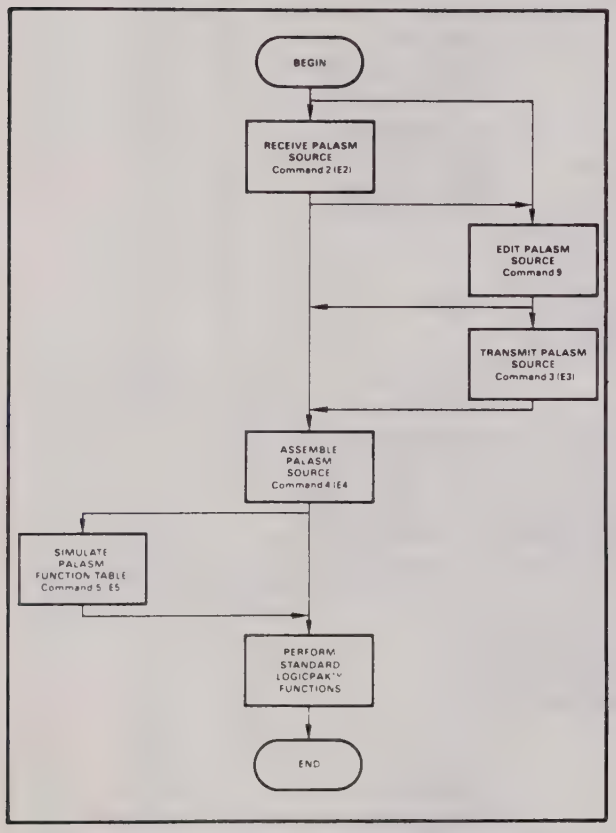

BEGIN
RECEIVE PALASM SOURCE Command 2 (E2)
EDIT PALASM SOURCE Command 9
TRANSMIT PALASM SOURCE Command 3 (E3)
ASSEMBLE PALASM SOURCE Command 4 (E4
SIMULATE PALASM FUNCTION TABLE Command 5 E5
PERFORM STANDARD LOGICPAK™ FUNCTIONS
END

Figure 3-1. PALASM Operational Overview

Before you begin data development with the PALASM system, we recommend that you become familiar with sections 1 and 2 of this manual. These sections provide the background information you need to understand the overall system operation and the concepts behind PALASM.

To begin operating the PLDS system with the PALASM design adapter installed, you must first set up the PLDS and terminal, applying the procedures described in sections 2.2 and 2.6. You may now apply power as described in section 3.2. At power-up, the PLDS performs a system self-test and then waits for a command from the programmer front panel. The terminal is inactive at this time. Many PALASM system operations are possible at this point from the front panel (as documented in sections 3.4 and 3.5), but to fully utilize the system you must have a terminal. To enter the terminal-based PALASM system, you must enter select code E1 from the front panel, as described in section 3.5.1. The top level command menu of figure 3-6 appears on the terminal display at this time, listing all the commands described in sections 3.4 and 3.5. The programmer display shows a static action symbol to indicate PALASM terminal operation.

NOTE

At any point during terminal operation, you may return to front panel control by entering ESC on the terminal, VERIFY On the Model 29 or 100 front panel, or KEYBD on the Model 19.

CAUTION

Before removing the PALASM design adapter, you MUST return to front panel operation as described above. The programmer executes firmware resident in the design adapter, and interruption may result in loss of RAM data.

Table 3-1. PLDS System Command Summary

MODULE OR ADAPTER	COMMAND TYPE	FROM FRONT PANEL	VIA TERMINAL	COMMAND DESCRIPTION		SEE SECTION
LogicPak™ (with any adapter)		--	0	Display menu		3.5.2
		E 1	--	Enable terminal mode		3.5.1
		--	1	Enter family code and pinout code		3.5.3
		E 5[a]	5[a]	Enter reject count option		3.5.4
		E 6	6	Enter verify option		3.5.5
		E 7	7	Enter security fuse option		3.5.6
		E 8	8	Set number of Logic Fingerprint™ test cycles		3.5.7
		E 9	8	Enter starting vector and test-sum		3.5.7
		--	8	Enter structured test vectors		3.5.7
				0	Display menu	3.5.7
				D	Delete current vector	3.5.7
				R	Repeat current vector	3.5.7
				U	Display previous vector	3.5.7
				#(N)	Go to vector (N)	3.5.7
				space	Move cursor right	3.5.7
				backspace[b]	Move cursor left	3.5.7
				return	Display next vector	3.5.7
				CTRL Z	Exit vector editor	3.5.7
		E A	A	Display fuse pattern		3.5.8
		E B	B	Receive JEDEC data		3.5.9
		E C	C	Transmit JEDEC data		3.5.9
		E D	D	Display sum-check of fuse data		3.5.9
		E E	--	Edit fuse by number		3.5.10
		--	E	Edit fuse pattern		3.5.10
				0	Display menu	3.5.10
				#(N)	Go to fuse (N)	3.5.10
				space	Move cursor right	3.5.10
				backspace[b]	Move cursor left	3.5.10
				return	Display next row	3.5.10
				CTRL Z	Exit fuse editor	3.5.10
		E F	F	Display configuration number		3.5.11
[a]Except with PALASM adapter.		C E	G	Set option attributes		3.5.12
[b]CTRL H is the same as backspace.		--	ESC	Exit terminal control before removing adapter		3.5.13
PALASM Design Adapter	Development	--	0	Display menu		
		--	1	Enter family and pinout codes		3.4.1
		E 2	2	Receive PALASM source		3.4.2
		E 3	3	Transmit PALASM source		3.4.3
		E 4	4	Assemble PALASM source		3.4.4
		E 5	5	Simulate function table		3.4.5
	Edit	--	9	Edit source		3.4.6
		--		0	Display menu	3.4.6
		--		B	Display line 1	3.4.6
		--		C	Change text	3.4.6
		--		D	Delete character	3.4.6
		--		E	Display to end	3.4.6
		--		I	Insert/enter text	3.4.6
		--		K	Delete current line	3.4.6
		--		L	Display 24 lines	3.4.6
		--		R(M)(N)	Repeat M lines after N	3.4.6
		--		U	Display previous line	3.4.6
		--		# (N)	Go to line N	3.4.6
		--		space bar	Move cursor/prompt right	3.4.6
		--		back space[b]	Move cursor/prompt left	3.4.6
		--		return	Display next line	3.4.6
		--		DEL/RUB	Delete characters (I mode)	3.4.6
		--		CTRL P	Purge all text	3.4.6
		--		CTRL Z	Exit editor, C or I mode	3.4.6
[b]CTRL H is the same as backspace.		--	ESC	Exit terminal control before removing adapter		3.4.6

NOTE: ESC (escape) returns control to programmer front panel.

Table 3-1. PLDS System Command Summary (Cont.)

MODULE OR ADAPTER	COMMAND TYPE	FROM FRONT PANEL	VIA TERMINAL	COMMAND DESCRIPTION	SEE SECTION
H&L Design Adapter	Development	--	Ø	Display menu	Refer to H&L Design Adapter Manual
		--	1	Enter family and pinout codes	
		E 2	2	Receive data (IFL format)[c]	
		E 3	3	Transmit data (IFL format)	
	Edit	--	4	Edit mode	
		--		G Enter gate number	
		--		P Enter product term number	
		--		T Enter transition term number	
		--		V Move cursor forward	
		--		V Move cursor backward	
		--		F Display next term	
		--		R Display last term	
		--		N Enter next field	
		--		I Insert term	
		--		D Delete term	
		--		C Clear term	
		--		X Deactivate term	
		--		E Display edit sub-menu	
		--		Ø Exit edit mode	
		--		1 Return to edit mode	
		--		2 Serial input (receive IFL format)[c]	
		--		3 Serial output (transmit IFL format)	
		--		4 List low-order terms	
		--		5 List high-order terms	
		--		6 Compress terms	
[c]Integrated fuse logic, Signetics ASCII		--		CTRL Z Exit edit mode	
		--	ESC	Exit terminal control before removing adapter	
All P/T Adapters	Device	--	1	Enter family code and pinout code	Refer to P/T Adapter Manuals
		Load	2	Load fuse data from device to RAM	
		Verify	3	Verify fuse data and perform functional test	
		Program	4	Program device with RAM data	

NOTE: ESC (escape) returns control to programmer front panel

3.1.1 DATA ENTRY

There are four basic ways to enter data into the PLDS with the PALASM design adapter installed:

- Single-key commands. The PLDS responds immediately to the command. This is typical of top-level commands, entering verify or security fuse options, etc.
- Number entry. When entering multiple-digit numbers in response to a prompt (e.g. line numbers in the source editor or fuse numbers in the fuse editor), the digits are entered in the order desired. Leading zeros and entry from the right side are handled automatically.
- Source editing. Data entry when editing the source file is described in section 3.4.6.
- Line editing. Many situations require that a line of arbitrary length be edited. Examples include the vector editor, the fuse editor, family and pinout code entry, and attribute selection. Line editing is accomplished in the following manner:
  1. The cursor may be moved back and forth along the line to be edited using the space and backspace keys.
  2. When the cursor is positioned over the character to be edited, pressing the key for the desired character will enter it into the line.
  3. Pressing RETURN at any time ends the editing session for a given line. If errors exist on the line, the incorrect line will be redisplayed for you to re-edit.
  4. Pressing CTRL Z at any time will terminate editing of a line without making any changes to the original.

3.2 POWER UP

NOTE

If the LogicPak™ with an adapter installed is not in the programmer before power is turned on, you will hear a beep until the LogicPak™ is installed.

When power is applied, the programmer will perform an automatic self-test routine (see LogicPak™ manual). When the self-test routine is complete, the programmer will signal its readiness (see your programmer manual).

To turn the programmer on:

1. Plug the AC power cord into the power outlet.
2. Lift the power switch up to the ON position (see figure 3-2).

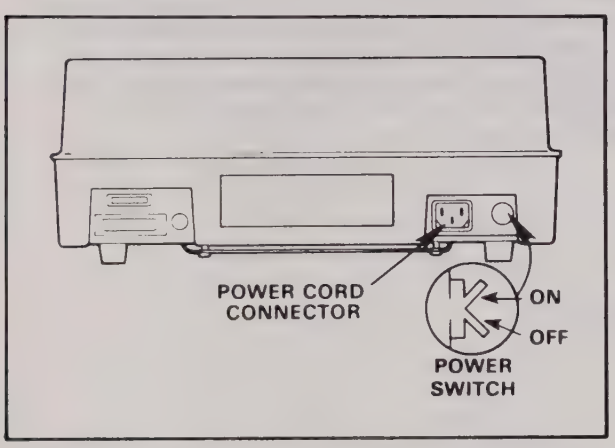


Figure 3-2. Programmer Power Switch Location

3.3 POWER DOWN

To turn the programmer power off:

1. Check to make sure the programmer is not in an operation process. If it is, wait until the operation is complete.
2. Push the power switch down to the OFF position (figure 3-2).

3.4 PALASM-SPECIFIC COMMANDS

The commands specific to operating the PLDS with the PALASM design adapter installed are described in this section. These commands are summarized in the PALASM design adapter section of table 3-1, and include commands for family and pinout code entry, receiving, transmitting, editing and assembling PALASM source, and simulating the PALASM function table.

3.4.1 FAMILY CODE AND PINOUT CODE SELECTION

Any device supported by the PALASM design adapter is specified by a combination of a two-digit family code and a two-digit pinout code. These codes are provided in appendix B of this manual. Since PALASM does not require the programming algorithm information provided by the different family codes, there is only one family code (E1) for all devices.

There are two ways to enter the family and pinout codes into PALASM:

- PALASM will automatically determine the correct family and pinout codes when a source assembly or a simulation is performed, since the device type is included in the source file.
- You may directly enter the family and pinout codes from terminal control (see section 3.5.3).

When PALASM operations are complete and you are ready to program a fuse pattern into a PAL, you will need to specify a new family code and possibly a new pinout code for the particular manufacturer's PAL you wish to program. This will be done during the normal course of device-related operations, and is described in the LogicPak™ and P/T adapter manuals.

3.4.2 RECEIVE PALASM SOURCE

This command prepares the programmer to receive a source file from a peripheral via the serial port. The incoming text will be placed in the source buffer, overwriting any existing source file.

CAUTION

The existing source file will be destroyed by receiving new PALASM source. To save your source file, refer to section 3.4.3.

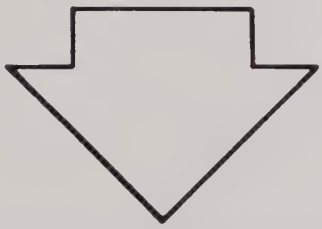
An example source file transmission and the resulting source buffer is shown in figure 3-3. The peripheral should transmit the entire source file and then terminate the transmission with either an ETX (ASCII hex 03) or a CTRL Z (ASCII hex 1A). If the peripheral cannot send a termination character, the command may be canceled from the programmer front panel by pressing SELECT to initiate the next operation.

The received text is stored sequentially in the source buffer as shown in figure 3-3. Only the printable ASCII characters and RETURN (ASCII hex 0D) are stored. A RETURN will be inserted in the source buffer if 80 printable characters are received without one. Lower-case letters (a...z) will be converted to upper case.

The incoming data should not contain the ESC character (ASCII hex 1B). The TAB character (ASCII hex 09) is ignored; it is not expanded.

TEXT TRANSMISSION

```
PAL12H6       PAL design specification (CR) (LF) (NULL) (NULL)
P01(CR) (LF)
BASIC GATES(CR) (LF)
DATA I/O(LF) (CR)
C D F G M N P Q I GND J K L R O H E B A VCC(CR) (LF)
(CR) (LF)
B = /A          ;INVERTER(CR) (LF) (ETX)
```



CONTENTS OF SOURCE BUFFER

HEX	ASCII
00 50 41 4C 31 32 48 36 20 20 20 20 20 20 50 41	.PAL12H6 PA
4C 20 44 45 53 49 47 4E 20 53 50 45 43 49 46 49	L DESIGN SPECIFI
43 41 54 49 4F 4E 20 0D 50 30 31 0D 42 41 53 49	CATION .P01.BASI
43 20 47 41 54 45 53 0D 44 41 54 41 20 49 2F 4F	C GATES.DATA I/O
0D 43 20 44 20 46 20 47 20 4D 20 4E 20 50 20 51	.C D F G M N P Q
20 49 20 47 4E 44 20 4A 20 4B 20 4C 20 52 20 4F	I GND J K L R O
20 48 20 45 20 42 20 41 20 56 43 43 0D 0D 42 20	H E B A VCC..B
3D 20 2F 41 20 20 20 20 20 3B 49 4E 56 45 52 54	= /A ;INVERT
45 52 0D FF FF 00 00 00 00 00 00 00 00 00 00 00	ER..............

Figure 3-3. Receive Source File

If the incoming text exceeds the buffer size (see table 3-2), a buffer overflow error will occur. In front panel

Table 3-2. Source Buffer Size

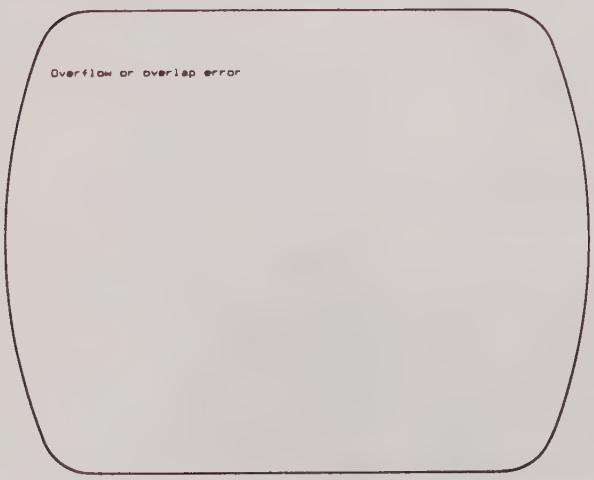
PROGRAMMER RAM	SOURCE BUFFER	START ADDRESS
64K	8K	2000
16K	8K	2000
8K	4K	1000
4K	2K	0800

operation an error 33 will be displayed and the operation will be aborted. Since the peripheral may still be sending data, an error 41 (overrun error) may occur at the start of the next serial port operation. During terminal operation an error message will be displayed and additional incoming data will be ignored until a termination character is received or a terminating command is issued.

If the source buffer overflows while receiving PALASM source:

29A Displays

Terminal Displays

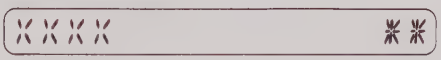


Front Panel Control

To prepare the PLDS to receive PALASM source, perform the following steps:

29A Displays

29A Displays



NOTE

[1] is the action symbol. XXXX represents the number of lines received.

Terminal Control

To prepare the PLDS for receiving PALASM source, enter a "2" from the top-level command mode:

Terminal Displays

NOTE

Data entered at this point from the terminal (including termination characters ETX and CTRL Z) is received as PALASM source. If a download link from a host computer is connected in parallel with the terminal, the downloaded data appears on the terminal display as it is being stored in the source buffer, and the termination character may be entered at the terminal.

3.4.3 TRANSMIT PALASM SOURCE

This command transmits the contents of the source buffer to the serial port. Each line is followed by a RETURN (ASCII hex 0D) and a line feed (ASCII hex 0A). The transmission is terminated with either an ETX (ASCII hex 03) or a CTRL Z (ASCII hex 1A). (See section 3.5.12 to select the termination character.)

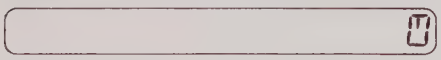
If the PLDS receives certain control characters during the course of the transmission, the transmission output will be affected as follows:

- CTRL S (DC1, ASCII hex 11): halt output.
- CTRL Q (DC2, ASCII hex 13): restart output after CTRL S.
- CTRL Y (ASCII hex 19): terminate transmission.
- ESC (ASCII hex 1B): terminate transmission and return to front panel control.

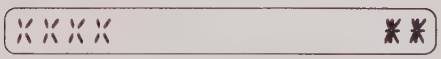
Front Panel Control

To transmit PALASM source from front panel control, perform the following steps:

29A Displays



29A Displays

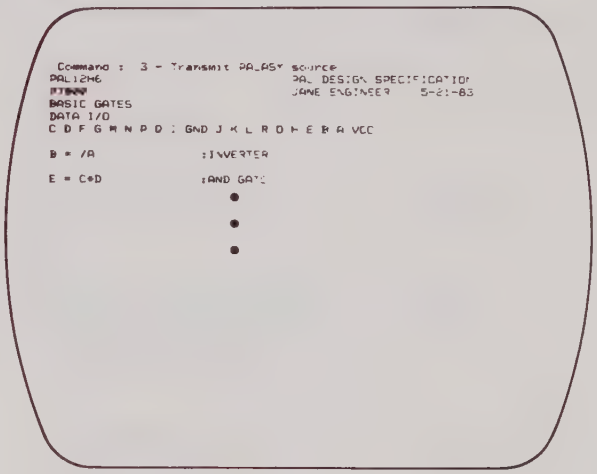


NOTE
[T] is action symbol. XXXX represents the number of lines transmitted.

Terminal Control

To transmit PALASM source from terminal control, enter a "3" in the top-level command mode:

Terminal Displays

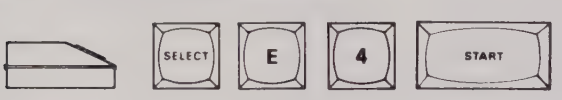


3.4.4 ASSEMBLE PALASM SOURCE

This command assembles (translates) the logic equations in the source buffer into a fuse pattern for programming a PAL. The PLDS PALASM is functionally equivalent to the FORTRAN version of PALASM provided by the logic device manufacturers. There is no user interaction involved with the assembly of a PALASM source file. Once the assembly is initiated it will run to completion, notifying you if any errors occurred.

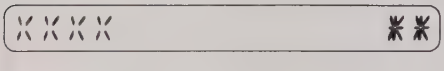
Front Panel Operation

To assemble a PALASM source file from front panel control, perform the following steps:



29A Displays

29A Displays

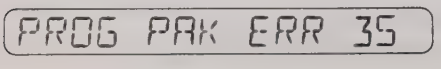


NOTE

[1] is action symbol. XXXX represents the number of fuses to be blown.

If errors occur during PALASM source assembly from the front panel mode, the following message appears:

29A Displays

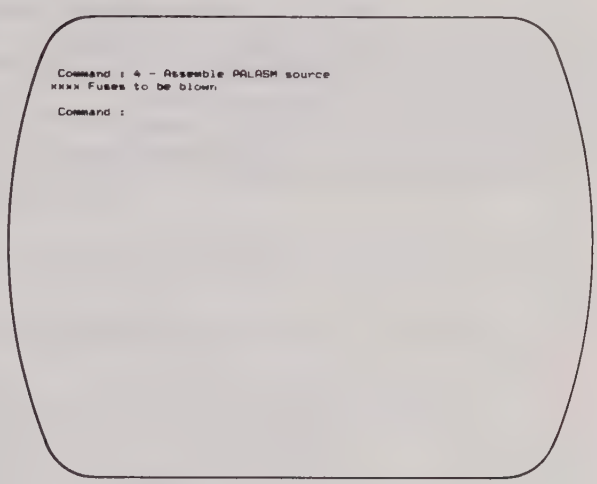


You must operate the PALASM assembler under terminal control to diagnose assembly errors more closely.

Terminal Operation

To assemble a PALASM source file under terminal control, press "4" from the top-level command mode:

Terminal Displays



When an error occurs during PALASM assembly in terminal control mode, a message with the following format is displayed:

```
Line 0007 Invalid Input Pin  <-- Line number and error
B = /AA          :INVERTER   <-- Incorrect source line
                             <-- Error indicator
```

The error indicator will point to the approximate location of the error.

NOTE

Some errors cannot be detected in the line in which they appear, but will cause an error in a later line.

The following source equations demonstrate an error which cannot be detected at the point at which it occurs:

```
B = /A +         :Inverter   <-- Line 0007
                             <-- Line 0008
E = C*D          :AND Gate   <-- Line 0009
```

The error is actually the "+" on line 7. Since PALASM ignores comments and allows equations to be continued on several lines, the error cannot be detected until line 9. Since "E" is an output-only pin the error will be reported as follows:

```
Line 0009  Invalid Input Pin
E = C*D            :AND Gate
```

If "E" was an input/output pin (such as found on the 16L8) the error would not be detected until the "=" on line 9. Note that PALASM cannot determine whether the designer had intended to leave out the "+" on line 7 or to enter a "+" instead of the "=" on line 9, either of which may have produced a correct source file.

Figure 3-4 shows a source file full of errors and the error diagnostics which result from assembling the file. Note that a fuse pattern is generated even when errors occur, but it will not be correct. Whenever errors occur during assembly, the message, "Source equation translation error," will appear as in figure 3-4.

```
0001 PAL12H6                              PAL DESIGN SPECIFICATION
0002 P7000                                DUMB ENGINEER    5-21-83
0003 BASIC GATES
0004 DATA I/O
0005 C D F G M N P Q I GND J K L R O H E B A VCC
0006
0007 A = /B               :INVERTER
0008
0009 E = CAT*D            :AND GATE
0010
0011 M := F + G           :OR GATE
0012
0013 L = /I + /J + /K     :NAND GATE
0014
0015 /O = M + N + A       :NOR GATE
0016
0017 R = P :+: Q          :EXCLUSIVE OR GATE
0019
0020 DESCRIPTION
0021 THIS EXAMPLE DEMONSTRATES MANY COMMON MISTEAKS.

-------------------------------------------------------------------------------

 Command :  4 - Assemble PALASM source
Line 0007 Invalid Output Pin
A = /B                    :INVERTER

Line 0009 Invalid Input Pin
E = CAT*D                 :AND GATE

Line 0011 Register Error
M := F + G                :OR GATE

Line 0015 Phase Error
/O = M + N + A            :NOR GATE

Line 0015 Too Many Terms Error
/O = M + N + A            :NOR GATE

Line 0017 XOR Error
R = P :+: Q               :EXCLUSIVE OR GATE

0207 Fuses to be blown

Source equation translation error

 Command :
```

Figure 3-4. Assembling a Source File With Errors

Table 3-3 lists all error messages which may be displayed by the assembler, and their meaning.

Table 3-3. PALASM Assembler Error Messages

ERROR MESSAGE	MEANING
Illegal Device Error	Line 1 does not start with a valid PAL type, i.e., PAL12H6. The number must start in column 4.
Pin List Not Found	No valid pin names found.
Illegal Character Error	A non-printable character was detected in the source buffer.
Pin List Error	Too few valid names in pin list. Pin list must start on line 5.
Invalid Output Pin	This pin is not a valid output.
IF( ) Error	The three-state conditional term is not allowed on this output. An "OR" was used in the product term.
Phase Error	The phase between the pin list and the output symbol is not correct for this device.
Register Error	An ":=" was used on a non-registered output or an "=" was used on a registered output.
Fixed Symbol Error	Error detected while evaluating fixed symbol. Fixed symbols are only allowed in PAL16A4s and PAL16X4s. See section 1.2.
Colon Error	Error in ":=", ":+:" or ":*:" symbol.
Equate Error	An "=" was found on the right side of the equation.
XOR Error	An ":+:" was found in the equation of a non-XOR output.
Parenthesis Error	An "(" or ")" was found while processing product term, only allowed in IF( ) and Fixed Symbols.
Too Many Terms Error	More product terms were found than are allowed for this output.
Invalid Input Pin	The pin name on the right side of equation is not a valid input or was not found in the line 5 pin list.

3.4.5 SIMULATE FUNCTION TABLE

This command uses the function table section of a source file (described in section 1.3.1) to simulate the operation of a PAL, based on the data in the fuse RAM. This involves comparing the desired states in the function table to an internal model of the selected PAL programmed with the pattern in fuse RAM. The simulator produces structured test vectors as a result of its operation.

The pattern in the fuse RAM may be the result of an assembly of source equations (section 3.4.4) or may be produced by any other method of loading fuse RAM. This includes receiving JEDEC-format fuse data (section 3.5.9), manually loading the fuse RAM using the fuse editor (section 3.5.10), or copying an actual device using a P/T adapter. The function table must exist in the source buffer; the simulator does not utilize structured test vectors already in RAM.

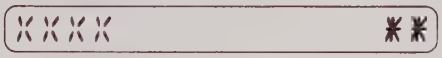
Be careful with the use of X ("don't care") on an input pin in your function table. The popular FORTRAN version of PALASM treats an X input as logic level 0. The PLDS PALASM simulator requires an L to produce a logic level 0. Any X inputs are not included in the evaluation of a product term. If all other inputs are TRUE the product term will be TRUE, therefore if all inputs are "don't care" the product will be TRUE. If all inputs are X in FORTRAN PALASM, the product term is evaluated as FALSE. The PLDS P/T adapters use the last defined state of the pin when X is specified in a structured test vector. In general, remember that "don't care" means you don't care what state is applied to the input.

Front Panel Operation

To simulate a function table against fuse RAM from the front panel, perform the following steps:

29A Displays

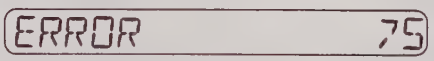
29A Displays



NOTE
XXXX represents the number of structured test vectors produced.

If errors are detected in the course of simulation from the front panel, the following message is displayed:

29A Displays



The simulator must be operated under terminal control to diagnose errors more closely.

Terminal Operation

To simulate the function table against fuse RAM from terminal control, press "5" from the top-level command mode:

Assuming that the Basic Gates source file of figure 1-6 has been entered into the source buffer and the corresponding fuse pattern exists in fuse RAM, simulation will proceed at the rate of approximately two vectors per second, producing the display below:

Terminal Displays

```
Command : 5 - Simulate function table
0001 XXXXXXXXXNXXXXXXH0N
0002 XXXXXXXXXNXXXXXXL1N
0003 00XXXXXXXNXXXXXXL0XN
0004 01XXXXXXXNXXXXXXL1XN


0020 XXXXXX00XNXXXLXXXXXN
0021 XXXXXX01XNXXXHXXXXXN
0022 XXXXXX10XNXXXHXXXXXN
0023 XXXXXX11XNXXXLXXXXXN

Command :
```

(Vectors 5 through 19 not shown)

If certain control characters are entered from the terminal during the course of simulation, the operation will be affected as follows:

- CTRL S (DC1, ASCII hex 11): halt simulation.
- CTRL Q (DC2, ASCII hex 13): restart simulation after CTRL S.
- CTRL Y (ASCII hex 19): terminate simulation.
- ESC (ASCII hex 1B): terminate simulation and return to front panel control.

Two types of errors are detected during simulation: syntax errors in the function table, and logical errors resulting from incorrect state definitions in the function table. The format of the display for each type of error is shown below.

Syntax errors:

```
Line 0027 Invalid Test Condition
XX   LHL   XXR     XXXX    XXX    XXX
           ^
```

Error indicator

Incorrect source line

Source line and error

The up arrow (^) points to the approximate location of the error. Any function table line with a syntax error will be ignored.

Logical errors:

```
0001 - L not found on pin 0018 - B
0001 XXXXXXXXXNXXXXXXL0N
```

When logical errors are detected, the error message is displayed immediately above the incorrect vector. The format of the error message is:

```
0001 - L not found on pin 0018 - B
```

Pin Name

Pin Number

Expected Condition

Vector Number

If the example errors shown above were to occur in the simulation of the Basic Gates example of figure 1-6, the display would show:

Terminal Displays

```
 Command :  5 - Simulate fuse data
@@@1 - L not found on pin @@18 - B
@@@1 XXXXXXXXXNXXXXXXL@N
@@@2 XXXXXXXXXNXXXXXXL1N
@@@3 @@XXXXXXXNXXXXXXLXXN
Line @@27 Invalid Test Condition
 XX   LHL    XXR    XXXX    XXX    XXX

@@@4 1@XXXXXXXNXXXXXLXXN


@@19 XXXXXX@@XNXXXLXXXXN
@@20 XXXXXX@1XNXXXHXXXXN
@@21 XXXXXX1@XNXXXHXXXXN
@@22 XXXXXX11XNXXXLXXXXN

Structured test verify error

 Command :
```

(Vectors 5 through 18 not shown)

Whenever errors occur during simulation from terminal control, the message, "Structured test verify error," is displayed at the end of simulation as shown above.

The error messages which may be displayed due to simulation errors are listed in table 3-4, along with their meaning.

Table 3-4. Simulator Error Messages

ERROR MESSAGE	MEANING
FUNCTION TABLE not found	The key word FUNCTION TABLE was not found.
Illegal Device Error	Line 1 does not start with a valid PAL type, i.e., PAL12H6.
Pin List not found	A valid symbol was not found on line 5.
Pin List Error	The pin list starting on line 5 does not have the proper number of valid symbols. The FUNCTION TABLE pin list has a symbol not found in the line 5 pin list. A dashed line was not found after the FUNCTION TABLE pin list.
Illegal Character Error	A non-printable character was found in the source buffer.
Invalid Test Condition	An invalid test condition was found. Only, C, X, F, Z, H, and L allowed. Too few conditions found before the end of the line or comment.

3.4.6 EDIT SOURCE

The source editor is a line editor tailored for generation and modification of PALASM source code. The editor operates on text stored in the source buffer. The source buffer may be loaded from a host development system as described in section 3.4.2, or text may be generated from an empty buffer using the source editor Insert mode described below.

Source editing may not be performed from the front panel. To begin editing source code from the terminal, press "9" from the top-level command mode:

Terminal Displays

The source editor command menu shown above appears when the editor is entered. Note that two types of commands are available: those that affect the display or cursor location, and those that actually edit text. The amount of source buffer memory presently filled also appears. Refer to table 3-2 for the maximum source buffer size.

The source editor is capable of editing lines longer than the 72 characters displayed on the terminal, up to 80 characters. This may be the case when you generate 80 character lines on a development system and download them. You may also add characters to a line in the Insert mode until the count of 80 is reached, at which point a new line is automatically created. You may edit beyond the right screen margin by moving the cursor to the right screen margin (where the cursor freezes) and then counting invisible character spaces to the desired editing point. However, shortening long lines using the D command is preferable.

The editor may be placed in any one of three modes at a given time: normal Command mode, Insert mode, or Change mode.

Editor Command Mode

In the Command mode, no marker appears at the cursor location and the editor will respond to any menu command at any point on a line of text. This is the default mode when the editor is entered. Table 3-5 lists the commands available in Command mode and describes their function. An example editing session is provided at the end of this section which illustrates many of the commands.

Table 3-5. Editor Commands

COMMAND	FUNCTION	DESCRIPTION
(Zero)	Display editor menu	Causes the editor menu to be redisplayed. The editor cursor remains on the same line. Source buffer usage is also displayed.
B	Beginning (line 0001)	The cursor is moved to the first character in the source buffer.
E	Display to end	All text from the current line to the end of text is sent to the display. Output may be temporarily halted and then restarted using CTRL S and CTRL Q, and the command may be terminated using CTRL Y.
L	List 24 lines	Displays the next 23 lines. Output may be temporarily halted and then restarted using CTRL S and CTRL Q, and the command may be terminated using CTRL Y.
U	Up (previous line)	Moves the cursor to the previous line (the line just "up" on the display).
RETURN	Go to next line	Moves the cursor to the start of the next line.
#(N)	Go to line N	When you press the "#" key, the editor will prompt you with "Go to line: " for the desired line number. The last four digits entered are accepted as the line number. Entering a line number greater than that of the last line in the source buffer will take you to the last line. Default is to remain on the same line.
Space	Move cursor right	Use the space bar to move the cursor to the right along a line of text. To insert characters at the end of an existing line, simply repeat spaces until the cursor stops (at one space past the final character) and then enter the Insert mode (described earlier in this section).
Backspace (CTRL H)	Move cursor left	Moves the cursor one character to the left along a line of text. This command operates in the Command, Insert, and Change modes. It is not possible to backspace beyond the first character on a line.
C	Change characters	The C command causes the editor to enter the Change mode as described earlier in this section.
D	Delete character	Causes the character currently at the cursor position to be deleted from the text. All characters to the right are moved one position left to fill the space.
I	Insert text	The I command causes the editor to enter the Insert mode as described earlier in this section.
K	Delete (Kill) current line	Deletes the entire line on which the cursor appears. You may issue the K command with the cursor at any point on the target line. After you delete a line, the same line number is displayed with the previous higher numbered line of text, since all lines have been moved down one to fill the space left by the deleted line.

Table 3-5. Editor Commands (Cont.)

COMMAND	FUNCTION	DESCRIPTION
R(M)(N)	Repeat (M) lines after (N)	The R command is used to copy (Repeat) a block of lines including the current line. The block of lines is actually inserted into the existing text, moving existing lines as required. This is particularly useful when several long equations of a similar form are being generated. When you give the R command, the editor prompts for the line count with "Repeat current line plus how many lines?". You may then enter a number from 0 to 99. Only the last two digits entered are accepted. 0 is the default if no line count is entered. If you enter a line count which exceeds the end of text, only the existing lines will be repeated. An error message will be displayed and no text will be moved if there is insufficient room in memory to repeat the specified lines. A second prompt is now issued ("After line") to accept the desired destination line for the repeat operation. The last four digits entered are accepted, with 0000 (repeat to beginning of buffer) being the default. A destination line number greater than the last line in the text will repeat at the end of text. An overlap error will occur if the destination line lies within the block of lines to be repeated. After executing the R command, the editor displays the original line number, with a different line of text if the destination line number was lower than the original line number (due to the extra lines now inserted).
Rubout/Delete	Delete character (I mode)	This command is invalid in the Command mode, but deletes the character just to the left of the cursor, in the Insert mode.
CTRL P	Purge all text	Entering a CTRL P will wipe out all text in the source buffer and display a blank line 0001.
CTRL Z	Exit I mode, C mode, or editor itself	CTRL Z is the general Exit command for all PLDS editors. In the source editor, it will exit back to the Command mode if in the Insert or Change modes, or from the source editor back to the top level command processor if in the Command mode.

Insert Mode

To enter the Insert mode from the Command mode, press an "I" at any point on a line of text:

Now a "<" marker appears at the cursor location, and any characters typed in will be inserted into the source text, moving other characters to the right to make space. Lower-case characters are converted to upper-case. While in the Insert mode, the only control characters available are Backspace (CTRL H) to move the insert point one character left, and Delete (or Rubout), to erase the character just left of the cursor.

If you insert text until the source buffer is filled, an "Overflow or Overlap error" message will be displayed. Each additional character entered at this point will cause the last character in the source buffer to be lost.

To exit the Insert mode back to the editor Command mode, enter CTRL Z:

NOTE

Editor response in the Insert mode may be slowed when editing long lines at lower baud rates. In this situation it is possible to garble the displayed line if characters are entered too rapidly.

EXAMPLE:

Original line 0001 THIS IS A_LINE

Enter Insert mode 0001 THIS IS A< LINE

Type "(space)" T E S T" 0001 THIS IS A TEST< LINE

Exit Insert mode 0001 THIS IS A TEST_LINE

Change Mode

To enter the Change mode from the Command mode, press "C" at any point on a line of text:

Now a "Λ" marker appears at the cursor, and any characters typed in are written over the existing text. Lower-case characters are converted to upper-case. The only control character available in Change mode is Backspace (CTRL H), which moves the cursor one character left.

To exit the Change mode back to the editor Command mode, enter CTRL Z:

If you reach the end of a line while in the Change mode, the editor automatically switches to Insert mode to allow you to continue entering text. The cursor marker also changes from "Λ" to "<".

EXAMPLE

Original line 0001 THIS IS A TEXTT LINE

Enter Change mode 0001 THIS IS A TΛXTT LINE

Type "E S T (space)" 0001 THIS IS A TEST ΛLINE

Exit Change mode 0001 THIS IS A TEST _LINE

Source Editor Sample Session

In the following sample editing session, the actual keys to be pressed are shown inside quotes and separated by spaces. Control keys (such as RETURN and the space bar) are shown lower-case and in parentheses. No distinction is made between command keys and text entry, since the context should be clear. For example, typing the phrase "TEST LINE" would be shown as:

Type "T E S T (space) L I N E"

1. Suppose that we have an original source file in the editor as shown below (note cursor on line 0007):

```
0001  PAL20X10                          PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                  L. BARRERE
0003  BRAND X COMPANY
0004  NOWHERE, USA
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007 _
```

2. We want to remove lines 0003 and 0004. First go to line 0003:

Type "# 3 (return)"

```
0001  PAL20X10                          PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                  L. BARRERE
0003  BRAND X COMPANY
0004  NOWHERE, USA
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007
Go to line:  0003
0003 _BRAND X COMPANY
```

3. To delete the current line,

Type "K"

```
0001  PAL20X10                          PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                  L. BARRERE
0003  BRAND X COMPANY
0004  NOWHERE, USA
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007
Go to line:  0003
0003  BRAND X COMPANY
0003 _NOWHERE, USA
```

4. Note that line 0003 is now the previous line 0004. Again delete the current line:

   Type "K"

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                        L. BARRERE
0003  BRAND X COMPANY
0004  NOWHERE, USA
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /DIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007
Go to line:  0003
0003  BRAND X COMPANY
0003  NOWHERE, USA
0003 _CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /DIN GND
```

5. The old lines 0003 and 0004 are now gone. To list the source buffer from the beginning:

   Type "B L"

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                        L. BARRERE
0003  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /DIN GND
0004  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0005 _
```

6. Now move the cursor to the end of line 0002. Repeat-space means to send spaces until the cursor stops automatically at the end of the line.

   Type "# 2 (return) (repeat-space)"

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                        L. BARRERE
0003  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /DIN GND
0004  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0005
Go to line:  0002
0002  6-BIT COUNTER                                        L. BARRERE_
```

7. Enter the Insert mode and type the correct lines 0003 and 0004:

   Type "I (return) D A T A I / O (return)
   R E D M O N D , (space) W A . "

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                        L. BARRERE
0003  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /DIN GND
0004  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0005
Go to line:  0002
0002  6-BIT COUNTER                                        L. BARRERE
0003  DATA I/O
0004  REDMOND, WA.↵
```

8. Now exit from the Insert mode back to the Command mode and list from the beginning to see what we have:

   Type "(CTRL Z) B L"

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                    L. BARRERE
0003  DATA I/O
0004  REDMOND, WA.
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007 _
```

9. Now a set of six equations may be generated using the R command. Type in the first equation using the Insert mode, and then place the cursor on the first line of the equation. The display should look like this:

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                    L. BARRERE
0003  DATA I/O
0004  REDMOND, WA.
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007
0008  /Q0 :=  /SET* _D*/D0                                  ;LOAD
0009       +  /SET*/_D*/Q0                                  ;HOLD
0010      :+: /SET*/_D* CNT* CIN* UP                        ;INCR
0011       +  /SET*/_D* CNT* CIN*/UP                        ;DECR
0012
0013
Go to line: 0008
0008 _/Q0 :=  /SET* LD*/D0                                 ;LOAD
```

10. Now repeat lines 0008 through 0012 after line 0012:

    Type "R 4 (return) 1 2 (return)"

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                    L. BARRERE
0003  DATA I/O
0004  REDMOND, WA.
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007
0008  /Q0 :=  /SET* LD*/D0                                  ;LOAD
0009       +  /SET*/LD*/Q0                                  ;HOLD
0010      :+: /SET*/LD* CNT* CIN* UP                        ;INCR
0011       +  /SET*/LD* CNT* CIN*/UP                        ;DECR
0012
0013
Go to line: 0008
0008  /Q0 :=  /SET* LD*/D0                                  ;LOAD
Repeat current line plus how many lines?  04
After line  0012
0008 _/Q0 :=  /SET* LD*/D0                                 ;LOAD
```

11. List the source from the beginning to see the block of lines we have just repeated:

    Type "B L"

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                    L. BARRERE
0003  DATA I/O
0004  REDMOND, WA.
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007
0008  /Q0 :=  /SET* LD*/D0                                  ;LOAD
0009       +  /SET*/LD*/Q0                                  ;HOLD
0010      :+: /SET*/LD* CNT* CIN* UP                        ;INCR
0011       +  /SET*/LD* CNT* CIN*/UP                        ;DECR
0012
0013  /Q0 :=  /SET* LD*/D0                                  ;LOAD
0014       +  /SET*/LD*/Q0                                  ;HOLD
0015      :+: /SET*/LD* CNT* CIN* UP                        ;INCR
0016       +  /SET*/LD* CNT* CIN*/UP                        ;DECR
0017
0018 _
```

12. The Repeat command may now be used twice more to quickly create six equation patterns which may then be modified to produce the required equations. The result may look like:

```
0001  PAL20X10                              PAL DESIGN SPECIFICATION
0002  6-BIT COUNTER                                    L. BARRERE
0003  DATA I/O
0004  REDMOND, WA.
0005  CLK D0 D1 D27 D8 D9 /LD /CNT /UP SET /CIN GND
0006  /OC Q9 Q8 Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 VCC
0007
0008  /Q0 :=  /SET* LD*/D0                                  ;LOAD
0009       +  /SET*/LD*/Q0                                  ;HOLD
0010      :+: /SET*/LD* CNT* CIN* UP                        ;INCR
0011       +  /SET*/LD* CNT* CIN*/UP                        ;DECR
0012
0013  /Q1 :=  /SET* LD*/D1                                  ;LOAD
0014       +  /SET*/LD*/Q1                                  ;HOLD
0015      :+: /SET*/LD* CNT* CIN* UP* Q0                    ;INCR
0016       +  /SET*/LD* CNT* CIN*/UP*/Q0                    ;DECR
0017
0018  /Q2 :=  /SET* LD*/D27                                 ;LOAD
0019       +  /SET*/LD*/Q2                                  ;HOLD
0020      :+: /SET*/LD* CNT* CIN* UP* Q0* Q1                ;INCR
0021       +  /SET*/LD* CNT* CIN*/UP*/Q0*/Q1                ;DECR
0022
0023  /Q3 :=  /SET* LD*/D27                                 ;LOAD
0024       +  /SET*/LD*/Q3                                  ;HOLD
0025      :+: /SET*/LD* CNT* CIN* UP* Q0* Q1* Q2            ;INCR
0026       +  /SET*/LD* CNT* CIN*/UP*/Q0*/Q1*/Q2            ;DECR
0027
0028  /Q4 :=  /SET* LD*/D27                                 ;LOAD
0029       +  /SET*/LD*/Q4                                  ;HOLD
0030      :+: /SET*/LD* CNT* CIN* UP* Q0* Q1* Q2* Q3        ;INCR
0031       +  /SET*/LD* CNT* CIN*/UP*/Q0*/Q1*/Q2*/Q3        ;DECR
0032
0033  /Q5 :=  /SET* LD*/D27                                 ;LOAD
0034       +  /SET*/LD*/Q5                                  ;HOLD
0035      :+: /SET*/LD* CNT* CIN* UP* Q0* Q1* Q2* Q3* Q4    ;INCR
0036       +  /SET*/LD* CNT* CIN*/UP*/Q0*/Q1*/Q2*/Q3*/Q4    ;DECR
0037 _
```

3.5 SYSTEM COMMANDS

In addition to the copy (load or program), verify, edit, and select functions described in the Operation Section of your programmer manual, the LogicPak™ offers numerous system commands that allow you to manipulate data and set parameters. System commands are accessed by entering a two-character select code from the programmer front panel or a one-character menu code from the terminal. Some commands will prompt for data entry. The operational overview (figure 3-1) will help you develop data and program a device using the system commands and programmer operations. Table 3-1 lists the select codes for Data I/O programmers to enter system commands from the programmer front panel and the menu codes for control from a terminal in terminal mode.

NOTE

The sequence explanations assume no operating errors. If these occur, the programmer signals with a beep and displays a two-digit error code in front panel mode or an error message in terminal mode. It also beeps once when an incorrect key is pressed. Error codes are explained in appendix B (table B-1) and in your programmer manual. Some errors will return you to the programmer front panel control from the terminal mode.

3.5.1 ENABLE TERMINAL MODE

Select code E1 transfers control of the PLDS to the terminal. After control is transferred, the 29A will display only its action symbol and the terminal will display the command menu (see figure 3-5). This command allows you to access data development and remote operations resident in the design adapters and remote operations using the P/T adapters.

See section 2.4 for terminal setup procedure.

3.5.2 DISPLAY COMMAND MENU

This command causes the PLDS to redisplay its command menu on the terminal as shown in figure 3-5.

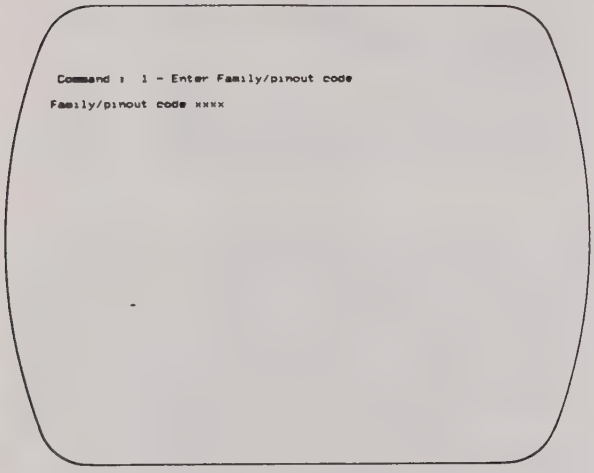
```
DATA I/O CORP. - PALASM Design Adapter - 303A-100-V02  (C) 1982.1983

      - GENERAL COMMANDS -                    - I/O COMMANDS -
0 - Display menu                        2 - Receive PALASM source
1 - Enter family/pinout code            3 - Transmit PALASM source
6 - Enter verify option                 B - Receive JEDEC data
7 - Enter security fuse option          C - Transmit JEDEC data
8 - Enter functional test data
F - Configuration number
G - Select attributes

    - SOURCE EQUATION COMMANDS -          - FUSE MAP COMMANDS -
4 - Assemble PALASM source              A - Display fuse pattern
5 - Simulate function table             D - Display fuse sumcheck
9 - Edit source                         E - Edit fuse pattern

NOTE - Always transmit an "ESC" before removing adapter
```

Figure 3-5. Main Command Menu

3.5.3 FAMILY CODE AND PINOUT CODE

Family and pinout codes may be determined automatically or entered from terminal control (see section 3.4.1). To enter the family and pinout codes from terminal control, press "1" in the top-level command mode:

Terminal Displays



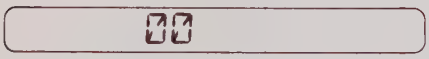
3.5.4 SET REJECT COUNT OPTION

This command allows you to select the number of programming pulses applied to the device fuses before the programmer rejects the device as unprogrammable. The default value of **0** selects the manufacturer's specified number of programming pulses. Refer to the adapter manual for specific entries to select optional reject values for single-pulse, military specifications, etc.

NOTE
The PALASM adapter does not provide this option.

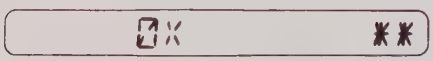
Front Panel Operation

29A Displays



To change the reject count to an optional value, enter the code number (X) specified in the adapter manual.

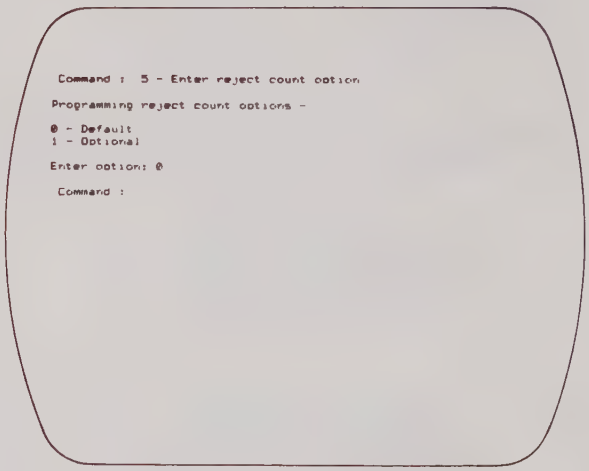
29A Displays



Terminal Operation

To select the reject count from the terminal enter a 5 from the command mode, then respond to the prompt with the count (X).

Terminal Displays



3.5.5 SELECT VERIFY OPTION

Three options are available for selecting verify and functional test routines. These routines are described in detail in sections 3.4.6 and 1.4.3 of the LogicPak™ manual.

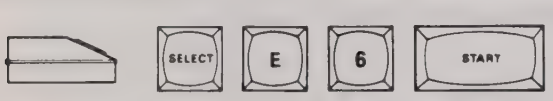
Options available are:

OPTIONS	DESCRIPTION
0	Default option. Perform fuse verify, followed by structured test (if test vectors are present in RAM), and Logic Fingerprint™ test (if one or more Logic Fingerprint™ test cycles are selected), in that order.
1	Perform fuse verify only.
2	Perform structured test and Logic Fingerprint™ test only, in that order. Do not perform fuse verify.

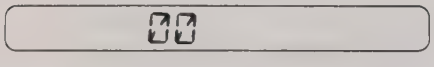
Option 0 (default) is the option used in normal operation. Option 1 checks the programming of the device fuses without checking device functionality. Use option 2 to functionally test devices with the security fuse blown. In addition, option 2 can be used to learn the Logic Fingerprint™ test of a device with the security fuse blown. Fuse data in RAM will be cleared during this operation. Programming cannot occur with option 2 selected.

Verify options must be entered from the programmer's keyboard or a terminal. The option will remain in effect until it is changed or until the unit is powered down. To reselect the default, key in option 0.

Front Panel Operation

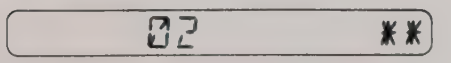


29A Displays



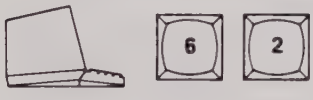
At this point, to select functional test **only** for example, do the steps which follow.

29A Displays

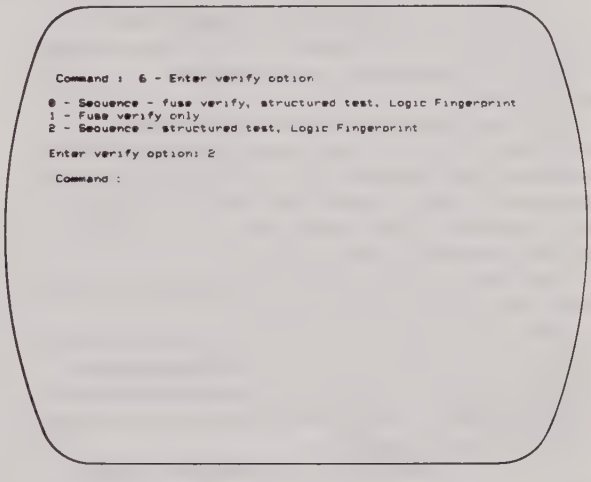


Terminal Operation

To enter the verify option from the terminal, enter 6 from the command mode, then respond to the prompt with the desired option. For example, to select functional test **only**:



Terminal Displays


Command : 6 - Enter verify option
0 - Sequence - fuse verify, structured test, Logic Fingerprint
1 - Fuse verify only
2 - Sequence - structured test, Logic Fingerprint
Enter verify option: 2
Command :

3.5.6 SELECT SECURITY FUSE OPTION

Some logic devices are equipped with protective fuses called security fuses. Once the security fuses are programmed, the fuse states in the logic array cannot be copied. Programming the security fuses makes it very difficult to pirate a device design.

The PLDS security fuse programming feature is a fail-safe function. You can either enable programming of the security fuse at all times, only allow programming when security fuse data are downloaded to the PLDS via the serial port, or disable programming completely, whether security fuse data are downloaded or not.

When the security fuse has been blown, a Logic Fingerprint™ test and structured test can still be performed, but a fuse verify operation is not possible (see section 3.5.5).

To enable programming of security fuses two conditions must be met: 1) the security fuse state in the programmer RAM must be 1 (or true), and 2) security fuse programming must be enabled. Once the security fuse option is selected, it will remain in effect until changed or until the programmer is turned off.

When security fuse data are entered into RAM in the JEDEC ASCII-logic format (see section 1.4.1 of the LogicPak™ manual), data in the **G** field indicate the state of the security fuse. The **G** field does not affect the enable state of the security fuse option; the enable state must be entered separately. This can be done before or after loading JEDEC ASCII-logic format data.

Security fuse states cannot be loaded from a master device.

CAUTION

Once the security fuse is blown, you can no longer verify the state of any fuse in the device. The process cannot be reversed; therefore, be certain that you want to program the security fuse before you activate this function. Attempting to re-program the device after the security fuse is blown will alter the original fuse pattern and render the device inoperative.

Front Panel Operation

To select a security fuse option from the front panel:

29A Displays

Security fuse select-code options are:

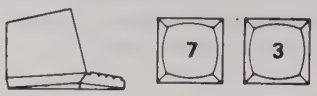
OPTION	DESCRIPTION
0	Default option. Disable programming and set the security fuse state in RAM to 0 (unprogrammed).
1	Disable programming, and set security fuse state in RAM to 1 (programmed).
2	Enable programming, and set security fuse state in RAM to 0. (Data downloaded in the JEDEC format can change the security fuse state to 1.)
3	Enable programming, and set security fuse state in RAM to 1. (Data downloaded in the JEDEC format can change the security fuse bit back to 0.)

For example, to enable security fuse programming and set security fuse state in RAM to 1 (option 3):

29A Displays

Terminal Operation

To enter the security fuse option from the terminal, enter 7 from the command mode, then respond to the prompt with the desired option. For example, to enable security fuse programming and set security fuse state in RAM to 1, do the following:



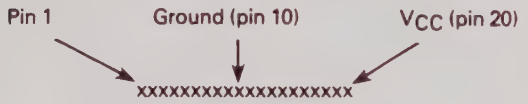
Terminal Displays

```
Command : 7 - Enter security fuse option

          SECURITY      SECURITY FUSE
OPTION    FUSE DATA      PROGRAMMING

   0 -------- 0 -------- disabled
   1 -------- 1 -------- disabled
   2 -------- 0 --------- enabled
   3 -------- 1 --------- enabled

Enter Security Fuse option: 3

Command :
```

3.5.7 ENTER FUNCTIONAL TEST DATA

Functional test data includes information for the Logic Fingerprint™ test and also the test vectors used by P/T adapters for testing of a programmed device. The Logic Fingerprint™ test information consists of three components:

- The number of test cycles to be performed during the Logic Fingerprint™ test (described later in this subsection). The default value is ØØ, which disables the Logic Fingerprint™ test.
- The Logic Fingerprint™ starting vector. This is an arbitrary binary sequence, each bit of which corresponds to a pin on the device under test. The starting vector format for a 20-pin device is shown below. Each "x" represents a "1" or a "0" to apply a logic high or logic low to the corresponding pin. Values entered for V_{CC} and ground affect the resulting Logic Fingerprint™ test signature, but have no effect on the device under test.



  The starting vector is used to initialize the Logic Fingerprint™, and is one of the components (along with the device type, number of test cycles, and programming pattern) which determines the resulting Logic Fingerprint™ test signature. Note that different Logic Fingerprint™ test signatures may result for a given logic design, depending on the choice of starting vector.
- The Logic Fingerprint™ test signature itself is the result of performing the Logic Fingerprint™ test as described later in this subsection.

Logic Fingerprint™ test data may be entered either from the front panel or from the terminal. From the front panel, the number of test cycles and starting vector for the Logic Fingerprint™ test may be entered, and the resulting Logic Fingerprint™ test signature may be viewed or entered. Structured test vectors may not be entered or edited from the front panel but only from a terminal or serial download. All functional test data may be entered from the terminal, including number of test cycles, starting vector, the Logic Fingerprint™ test signature itself, and the test vectors.

NOTE

If a value is entered for the Logic Fingerprint™ test signature, it should be either ØØØØØØØØ or a known-good value corresponding to the number of test cycles, starting vector, device, and fuse patterns under test. A value of ØØØØØØØØ will cause the LogicPak™ to "learn" the correct Logic Fingerprint™ test signature when a Load, Program, or Verify operation is performed (see section 3.4 of the LogicPak™ manual for details). When in Load, the correct Logic Fingerprint™ signature will be learned independent of the value entered.

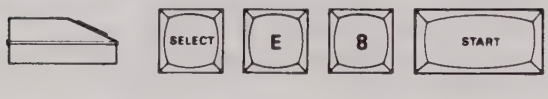
If "Device Selection Error" (Prog Pak 30) appears when you select functional test data, you must specify family code and pinout code to define the vector width.

In the subsections which follow, functional test data will be entered to test the Basic Gates design example (see figures 1-5 and 1-6).

Front Panel Operation

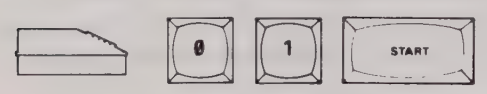
From the front panel, the number of test cycles and the Logic Fingerprint™ starting vector may be entered, and the Logic Fingerprint™ test signature may be viewed or entered.

Set Number of Logic Fingerprint™ Test Cycles.

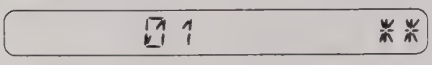


29A Displays

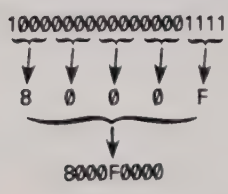
For example, to enable one cycle of testing,



29A Displays



Logic Fingerprint™ and Starting Vector. The starting vector must be converted from the binary form to hexadecimal for entry from the front panel. For our Basic Gates example, we will choose an arbitrary test vector as shown:



Starting vector (binary)

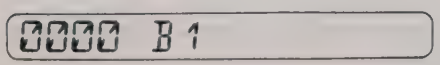
(hexadecimal)

Starting vector (hexadecimal)

The unused portion of the 32-bit vector is assumed to be zeroes and must be included in the hexadecimal vector entry.

For example:

29A Displays

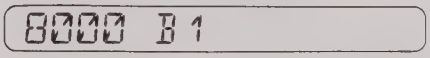


NOTE

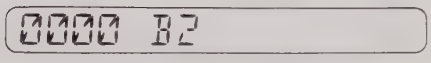
The eight-character starting vector is entered into the programmer in two fields. B1 identifies the first field.

To enter the first four hexadecimal digits,

29A Displays



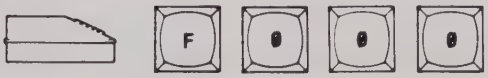
29A Displays



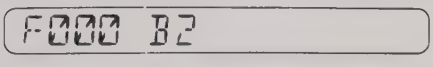
NOTE

B2 represents the second field.

Enter the remaining hexadecimal digit by pressing



29A Displays



The zeroes are ignored, but are needed to correctly position the "F."

This vector, when applied to the Basic Gates example, produces the Logic Fingerprint™ test signature:

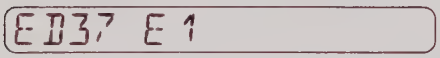
ED37A9E4 (hexadecimal)

This value may be viewed or entered at this time by pressing START:

1.

Enter the first four characters of the Logic Fingerprint™ test signature if desired.

29A Displays



The first four characters are displayed as the E1 field. The last four characters (E2 field) may be viewed or entered by pressing START again:

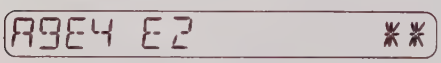
2.

Enter the next four characters if desired.

29A Displays

3.

29A Displays



Terminal Operation

Entering an "8" from the Command mode allows you to enter functional test data and begin vector editing from the terminal.

The functional test data may be entered in response to three prompts (see figure 3-6).

```
Command : 8 - Enter functional test data

Cycles for Fingerprint: xx
Fingerprint starting vector: xxxxxxxxxxxxxxxxxxxx
Fingerprint: xxxxxxxx

Note: x represents current values
```

Figure 3-6. Prompts for Entering Functional Test Data

As each prompt appears, you may modify the current values (represented by x's in figure 3-6) using the following steps:

1. Move the cursor forward (using the spacebar) and backward (using the backspace) along the displayed value until it is positioned over the symbol to be changed.
2. Type the desired symbol.
3. Enter RETURN or CTRL Z at any point to move to the next prompt.
4. CTRL Z is used to exit the functional test entry mode.

For our test example, the values shown in figure 3-7 should be entered.

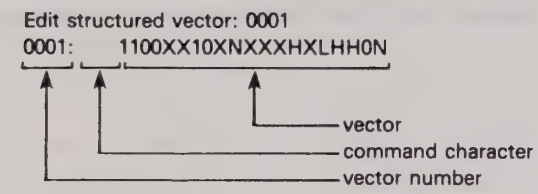
```
Command : 8 - Enter functional test data
Cycles for Fingerprint: 01
Fingerprint starting vector: 10000000000000001111
Fingerprint: ED37A9E4

          - DISPLAY -                        - EDITING COMMANDS -
0 ------------ Display menu          D ------- Delete (Kill) current vector
Return -------- Go to next vector     R ------- Repeat current vector
U ------------ Up (previous vector)   CTRL Z -- Exit vector editor
#(N) ---------- Go to vector (N)
Space --------- Move cursor right
BKSP (CTRL H) - Move cursor left
CTRL Z -------- Exit vector editor

Edit structured vector:  0000
0001:  1100XX10XNXXXHXLHH0N
0002:  --------------------
```

Figure 3-7. Entering Functional Test Data

Vector Editing. Vectors are created by downloading JEDEC 'V' fields, simulating a source file containing a function table, or by using the vector editor.

When the Logic Fingerprint™ test information has been entered (or skipped by entering RETURNs), the vector editor menu appears (see figure 3-7), and a prompt appears for the vector number to be edited. The default vector is 0001, as shown in figure 3-7.

The vector editor is a fixed-format line editor with the first column of the displayed line reserved for command characters as shown below.



A character entered in the first column (normally blank) is interpreted as a command and acted upon immediately; otherwise vector editing is not processed until a RETURN is entered (at any point on the line). The command characters recognized in the first column are 0, U, #, K, and R; see table 3-6 for command character definitions.

During operation, the vector editor copies the selected vector to a temporary buffer where all editing changes are made. Then, when a RETURN command is entered, the temporary buffer is examined for legal characters before copying back to vector memory. You are not allowed to proceed to another vector until all characters are legal in the current vector. Typing a CTRL Z to exit the vector editor will leave the selected vector in its original state.

Table 3-6. Vector Editor Command Characters

COMMAND	DESCRIPTION	ACTIVITY
0 (zero)	Display menu	Redisplays menu and restarts editing on the same vector.
U	Up (previous vector)	Moves editing to the next lower vector number (the vector one 'up' on the screen).
#(N)	Go to vector (N)	Entering a '#' in the command column causes the vector editor to prompt for the desired vector number (default = 0001). Entering a vector number greater than the last vector will move you to the last vector.
D	Delete current vector	Current vector is deleted, and all higher vectors moved down one. Current vector number is redisplayed with new vector.
R	Repeat current vector	Creates a copy of the current vector immediately following the current vector. The copy is displayed, with its vector number (one greater than the original). This command may be given for any vector, and existing vectors will be moved to accommodate the new copy.

An "empty vector" is represented by a dash in all pin positions. This will appear as the first vector in an empty vector editor buffer, or as one past the last vector where data are present in memory. All vectors are numbered lower than the empty vector.

To edit a vector, follow the steps below.

1. Move the cursor forward (using the spacebar) and backward (using the backspace) along the displayed vector until it is positioned over the test condition to be changed.
2. Type the desired test condition to enter it into the vector image; the allowable test conditions are 0-9, X, N, F, H, L, Z, C, and K (see table 3-7 for test condition definition).

Table 3-7. Vector Symbol Definition

VECTOR SYMBOL	DEFINITION
0	Drive input low
1	Drive input high
2-9	Drive input to supervoltage #2-9
C	Drive input low, high, low
K	Drive input high, low, high
N	Power pins and outputs not tested
L	Test output low
H	Test output high
Z	Test output for high impedance
F	Float input or output
X	Ignore input or output (not defined in JEDEC format)

NOTE

"X" is not defined in the JEDEC format. The "X" is treated as an "N" for outputs and leaves an input at its previously defined state.

Test conditions 2 through 9 specify non-TTL levels (super voltages) that access special device features. A device may be damaged by improper use of super voltages.

3. Enter RETURN or CTRL Z at any point to move to the next vector or to exit the vector editor.

3.5.8 DISPLAY FUSE PATTERN

This command transmits the fuse pattern in the programmer data RAM to the serial port. The fuse states may be shown as a series of "1"s and "0"s or a series of "-"s and "X"s; see section 3.5.12 on selecting characters. The "1"; or "-" represents a high resistance fuse, "blown" in a fuse link device. The "0" or "X" represents a low resistance or "intact" fuse. Each fuse can be identified by a decimal fuse number as shown in figure 3-8. The fuse states are arranged in a matrix that corresponds to the logic diagram of the device (figure 3-9). This is useful for comparing or copying a displayed fuse pattern to the device logic diagram. Logic diagrams and fuse number charts for all supported devices are in the appendix of the programming/testing adapter manuals.

```
Command : A - Display fuse pattern

      00          10          20
0000  -------X--  ----------  ----
0024  XXXXXXXXXX  XXXXXXXXXX  XXXX
0048  XXXXXXXXXX  XXXXXXXXXX  XXXX
0072  XXXXXXXXXX  XXXXXXXXXX  XXXX
0096  X-X-------  ----------  ----
0120  XXXXXXXXXX  XXXXXXXXXX  XXXX
0144  ----X-----  ----------  ----
0168  --------X-  ----------  ----
0192  ----------  -X-X------  ----
0216  XXXXXXXXXX  XXXXXXXXXX  XXXX
0240  ----------  ----X--X--  ----
0264  ----------  -----XX---  ----
0288  ----------  ----------  -X--
0312  ----------  ----------  ---X
0336  ----------  ---------X  ----
0360  XXXXXXXXXX  XXXXXXXXXX  XXXX
Sumcheck 1BB9

Command :

NOTE: - = open
      X = intact

Fuse Number = First Fuse Number + Increment
```

Figure 3-8. Complete Fuse Pattern

NOTE

Sending certain control characters to the PLDS during the course of fuse pattern display will affect the display. The output may be stopped by sending a CTRL S (DC1 or ASCII 11 hex) and then restarted by sending a CTRL Q (DC3 or ASCII 13 hex).

A CTRL Y (ASCII 19 hex) will terminate the transmission and return to the terminal or front panel operation.

An ESC character (ASCII 1B hex) will terminate the transmission and return to front panel operation.

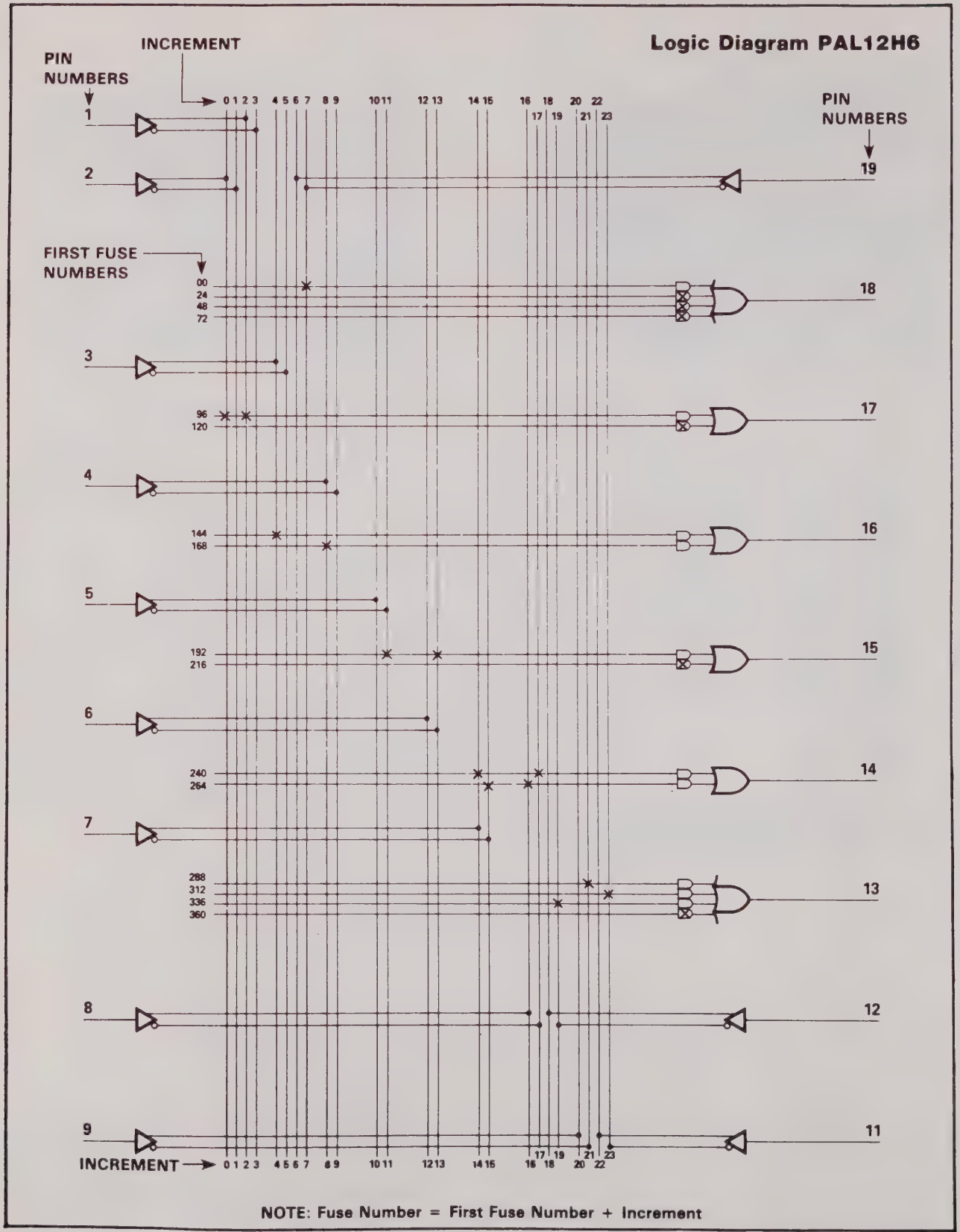


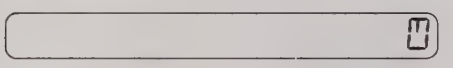
Figure 3-9. Logic Diagram for Basic Gates Example

The last character of the fuse pattern transmission is either CTRL C (ETX or ASCII 03) or a CTRL Z (ASCII 1A hex). (See section 3.5.9 on selecting the termination character.)

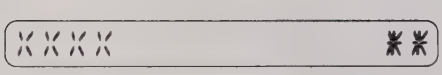
Front Panel Control

To display the fuse pattern from front panel control, follow these steps:

29A Displays



29A Displays



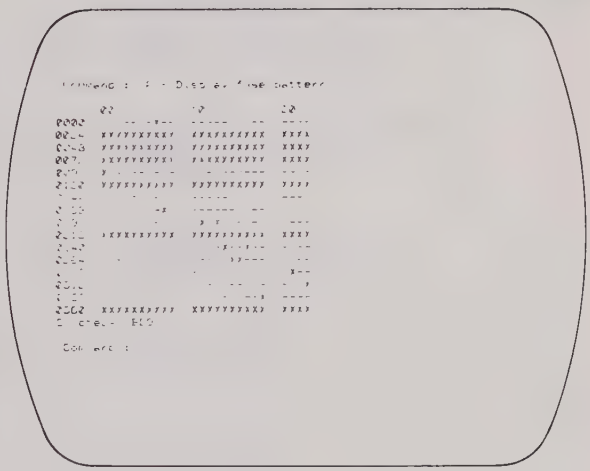
NOTE

□ is the action symbol. XXXX represents the fuse array checksum.

Terminal Control

To display the fuse pattern from the terminal command mode, enter an "A":

Terminal Displays



3.5.9 JEDEC FORMAT DATA EXCHANGE

Fuse data, test vectors, and the Logic Fingerprint™ test signature are transmitted between the host computer and the PLDS in the JEDEC format. The JEDEC format is described in detail in appendix A. A brief overview of the format is provided in this section, and shown in figure 3-11. Figure 3-10 shows an example JEDEC transmission and its components.

The transmission consists of a start-of-text (STX) character, the various fields, an end-of-text (ETX) character, and a transmission checksum as shown in figure 3-11.

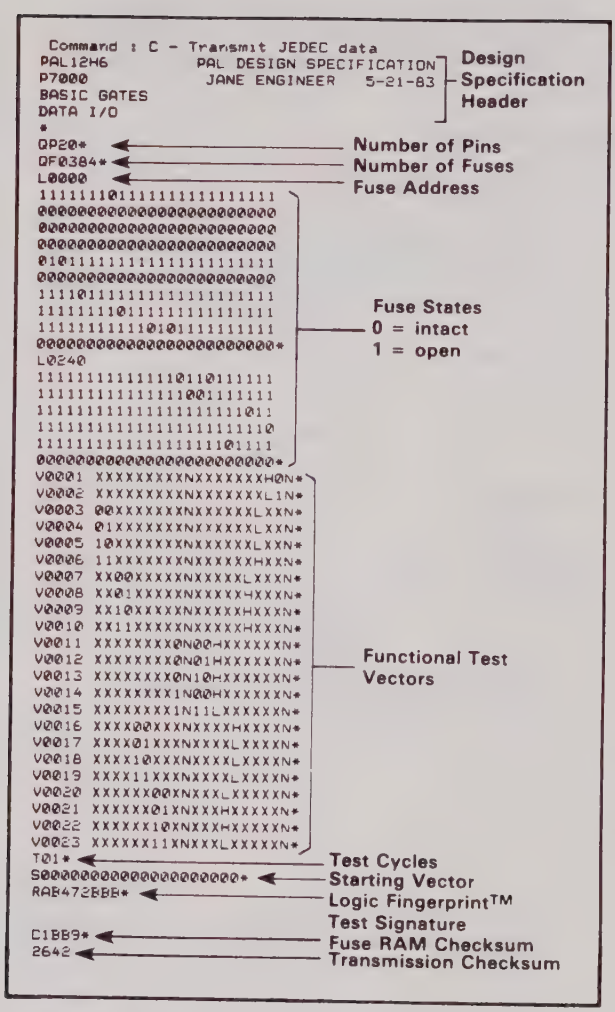


Figure 3-10. JEDEC Transmission—Basic Gates Example

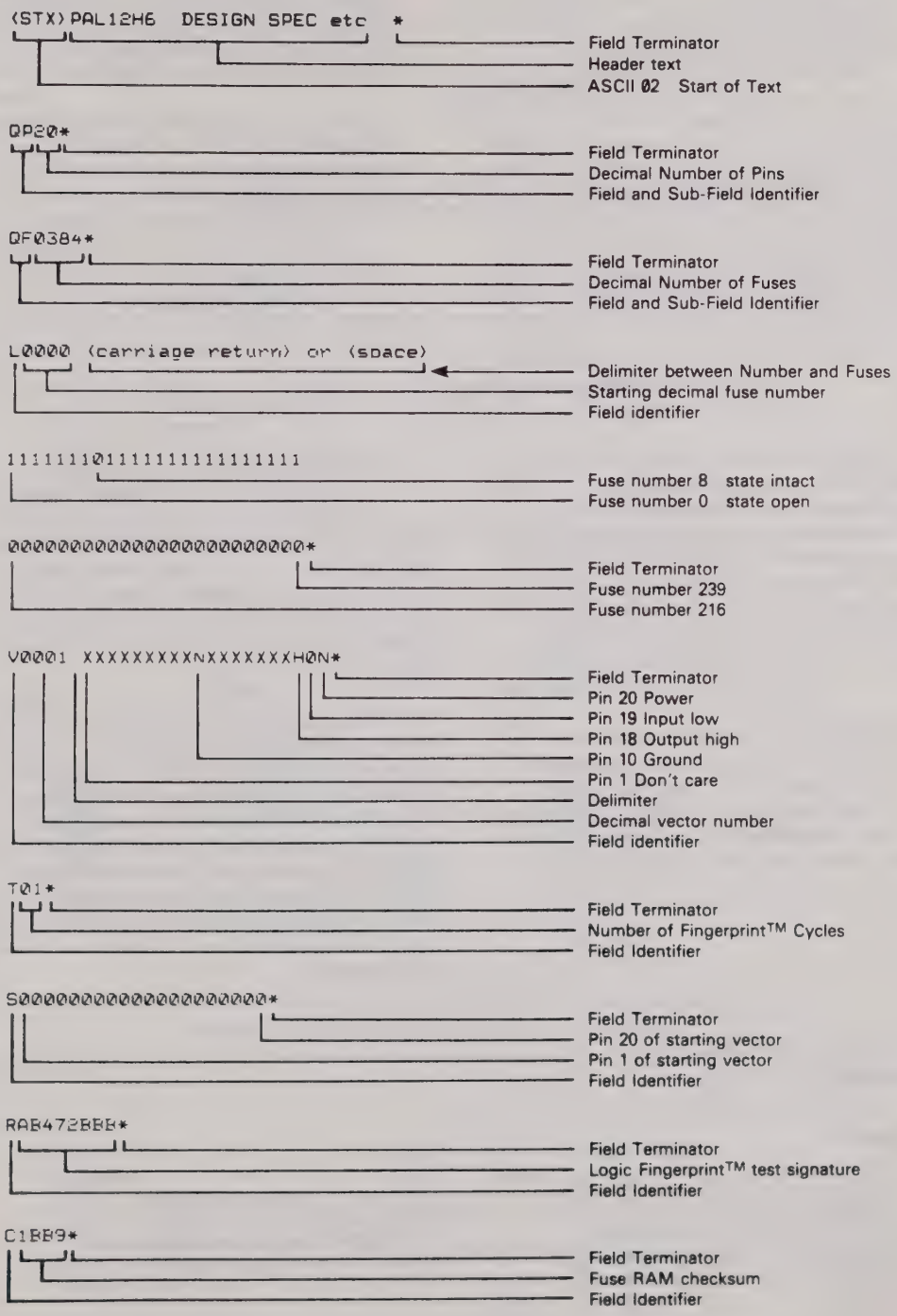


Figure 3-11. JEDEC Format (Breakdown of Figure 3-10)

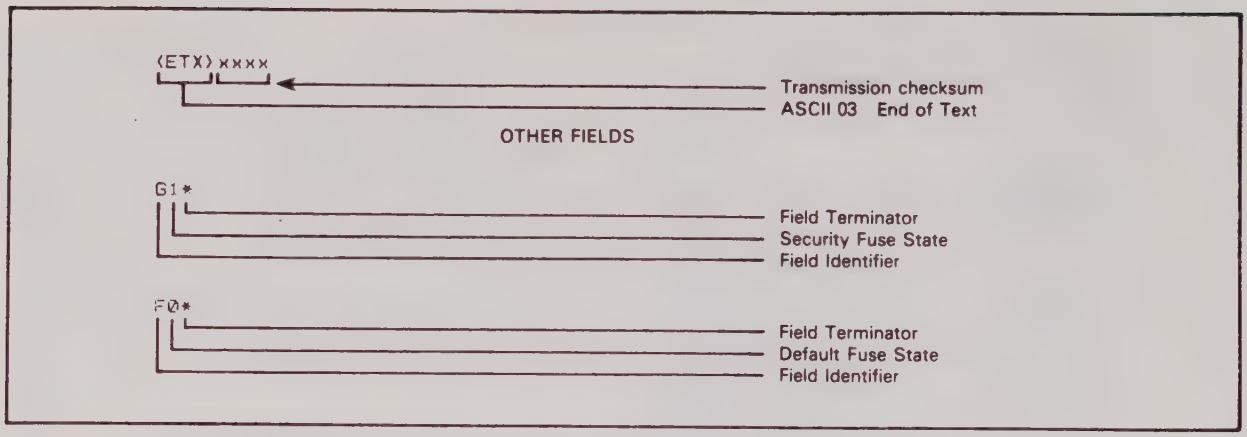


Figure 3-11. JEDEC Format (Breakdown of Figure 3-10) (Cont.)

The transmission checksum is the 16-bit sum of all ASCII characters transmitted between and including the STX and ETX. The parity bit is excluded in the calculation (see figure 3-12). The transmission checksum computed by the PLDS may be found by examining data RAM addresses 405 and 406, using the programmer's EDIT mode (discussed later in this subsection). Some computer operating systems do not allow a user to control what characters are sent, especially at the end of a line. The transmission checksum may be disabled in this case by sending a dummy checksum of "0000".

In general each field in the format starts with an identifier, is followed by the information, and is terminated with an asterisk. For example, "T01*" sets the number of Logic Fingerprint™ test cycles to 1. The design specification header does not have an identifier and must be the first field in the transmission, immediately following the STX.

Fuse information is specified by the "QF", "F", "L", and "C" fields. The "QF", "F", and "C" fields are optional.

The "QF" field sets the maximum allowable fuse number; this will override the value set by the family code and pinout code. The "F" field sets the default fuse value. An "F0*" fills the fuse RAM with 0s, and an "F1*" fills the fuse RAM with 1s. This operation takes a significant amount of time and can lead to an input buffer overflow at high baud rates.

The "L" field starts with a decimal fuse number and is followed by a stream of fuse states (1 or 0). The fuse number may include leading zeroes (i.e. "L12" and "L0012" are the same). A "space" and/or a "carriage return" must separate the fuse number from the fuse states. The stream of fuse states can be as long as desired (up to the maximum allowable fuse number). The fuse data for an entire device, for example, could be sent in one "L" field starting at zero and continuing for all fuses in the device. Spaces and carriage returns may be inserted to make the stream more readable. Each "L" field must be terminated with an asterisk.

Example 1:

```
(STX)*(ETX)002F          02 + 2A + 03 = 2F
```

Example 2:

```
random text (return)(line feed)                                    = 0000
(STX)TEST*(return)(line feed)          02+54+45+53+54+2A+0D+0A = 0183
DF0384*(return)(line feed)          51+46+30+33+38+34+2A+0D+0A = 01A7
F0*  (return)(line feed)                  46+30+2A+20+20+0D+0A = 00F7
L10 101*(return)(line feed)  4C+31+30+20+31+30+31+2A+0D+0A = 01A0
(ETX)05C4 (return) other random text                       03 = 0003
                                                                ----
                                                                05C4
```

Figure 3-12. Computing the Transmission Checksum

The "C" field is the sum-check of the entire fuse RAM (fuse number 0 to maximum fuse number for the selected device), not just the fuse states sent. See figure 3-13. (The JEDEC term "Fuse Checksum" is the same as Data I/O's term "sum-check".)

The structured test vector information is specified by the "QP", "P", and "V" fields. The "QP" field defines the number of pins on the device and overrides the value set by the family code and pinout code. The "V" field starts with a vector number, is followed by a space, then by a series of test conditions for each pin, then is terminated with an asterisk. The test conditions are normally sent in pin number order, however the "P" field can specify a different sequence. The PLDS JEDEC translator does not validate the test conditions in the vectors (see table 3-7 for the presently defined test conditions). The super-voltage test conditions (2 through 9) are used to apply non-TTL levels to certain pins to access special test features. A device could be damaged by improper use of super-voltages.

The Logic Fingerprint™ Test information is specified by the "T", "S", and "R" fields. The "T" field defines the number of test cycles to be performed. The legal values are 0 to 99. The "S" field defines the starting vector with a series of 1s and 0s for each pin. The "R" field defines the 8 digit hexadecimal Logic Fingerprint™ test signature.

The "G" field defines the security fuse state.

The "D" field is not sent by new versions of the PLDS JEDEC translator. It has been replaced by the "QF" and "QP" fields and the manual setting of family codes and pinout codes.

```
(STX)*F0*L0000
01001110 00001000 11110000 11111111 01010001*
C021A*
(ETX)0000
```

The F0* cleared all the fuse RAM to 0. The L field transmitted 40 fuse states starting at 0.

```
Fuse number  00 01 02 03 04 05 06 07   08 09 10 11 12 13 14 15   16
State         0  1  0  0  1  1  1  0    0  0  0  0  1  0  0  0    1


                  MSB              LSB

                   7 6 5 4 3 2 1 0

   0000            0 1 1 1 0 0 1 0          72
   0008            0 0 0 1 0 0 0 0          10
   0016            0 0 0 0 1 1 1 1          0F
   0024            1 1 1 1 1 1 1 1          FF
   0032            1 0 0 0 1 0 1 0          8A
   0040            0 0 0 0 0 0 0 0          00
   0048            0 0 0 0 0 0 0 0          00
   xxxx            0 0 0 0 0 0 0 0          00
                                           ----
                                           021A
```

Figure 3-13. Computing the Fuse RAM Checksum

Transmit JEDEC Data

This command transmits the contents of the fuse and vector RAM to the serial port in the JEDEC format (see appendix A).

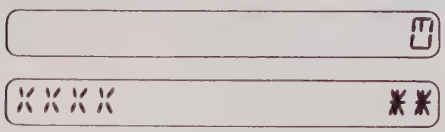
The following characteristics apply to JEDEC transmission.

- The output may be halted by sending a CTRL S (DC1 or ASCII 11 hex) and restarted by sending a CTRL Q (DC3 or ASCII 13 hex).
- An ESCAPE character (ASCII 1B hex) will abort the transmission and return to the programmer front panel operation.
- A CTRL Y (ASCII 19 hex) will terminate the transmission and return to the terminal or programmer front panel operation.
- The Logic Fingerprint™ test fields (S, R, and T) are not sent if the number of cycles is Ø.
- The "G" field is sent only if security fuse data is a "1."
- The fuse checksum (C field) is the 16-bit sum of all fuse states (*i.e.*, from fuse Ø to the fuse limit for the device). See figure 3-13.

Front Panel Control

To transmit JEDEC data, follow the steps below.

29A Displays



NOTE

[U] is the action symbol. XXXX represents the fuse array sum-check.

Terminal Control

To transmit JEDEC data from the terminal mode, enter a "C" from the Command mode:

See figure 3-10 for the terminal display of the Basic Gates design example data.

Receive JEDEC Data

This command prepares the programmer to receive fuse and vector data from a peripheral device via the serial port. A translator converts the JEDEC format data (see appendix A) to the memory image required by the PLDS.

NOTE

The D field is ignored by the translator. The correct family code and pinout code must be entered before receiving JEDEC data. See table B-1 in appendix B for correct family codes and pinout codes.

There are three types of errors that may be caused by receiving improper data in the JEDEC format (see table 3-8). You may determine the field in which the error

Table 3-8. Translation Input Errors

ERROR	DESCRIPTION	POSSIBLE FIELDS
82	SUMCHK ERR	Transmission check-sum
84	INVALID DATA	ETX F L S V
91	I/O FORM ERR	C G L P R T V

occurred by examining data RAM location 0408; the ASCII value (hexadecimal) of the field is stored here (see table 3-9). More information about the possible cause of the error may be found in table 3-10.

Table 3-9. ASCII Values of Field Identifiers

FIELD IDENTIFIER	ASCII VALUE
(ETX)	03
C	43
F	46
G	47
L	4C
P	50
R	52
S	53
T	54
V	56

To examine the data RAM location 0408, perform these steps.

29A Displays

29A Displays

NOTE
XXXX is the current address.
XX is the field identifier in hexadecimal.

The transmission checksum computed by the PLDS may be found by examining data RAM locations 405 and 406 in a similar fashion.

Table 3-10. Translator Input Error Codes

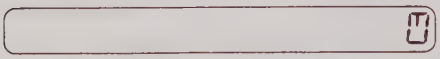
ERROR	DISPLAY	FIELD*	POSSIBLE CAUSE
82	SUMCHK ERR	ETX	Transmission checksum of all ASCII characters does not match the computed value.
84	INVALID DATA	ETX	Fuse sum-check does not match computed sum-check. The comparison is not made until the transmission is complete, so the field is stored as ETX rather than C. The sum-check includes the entire fuse RAM as defined by the family and pinout code, not just the fuse states sent.
		F	Invalid character in field. Only "1" and "0" are allowed.
		L	A space or carriage return did not follow the fuse number.
		L	An invalid character was in the fuse state field. Only "1" and "0" are allowed. Spaces, line feeds, and carriage returns are ignored.
		S	Invalid character in field. Only "1", "0", and "N" are allowed.
		V	Too few or too many test conditions.
91	I/O FORM ERROR	C	Invalid character in field, must be 4 digit hexadecimal number.
		G	Invalid character in field. Only "1" or "0" are allowed.
		L	Fuse number exceeds fuse limit for device or invalid fuse number (must be decimal number).
		P	Too few or too many pins or invalid pin number for device.
		T	Test cycles greater than 99.
		R	Invalid character in field: must be 8 digit hexadecimal number.

*From RAM Addr. 0408

Front Panel Control

To receive JEDEC data from the front panel mode perform the steps which follow.

29A Displays



29A Displays

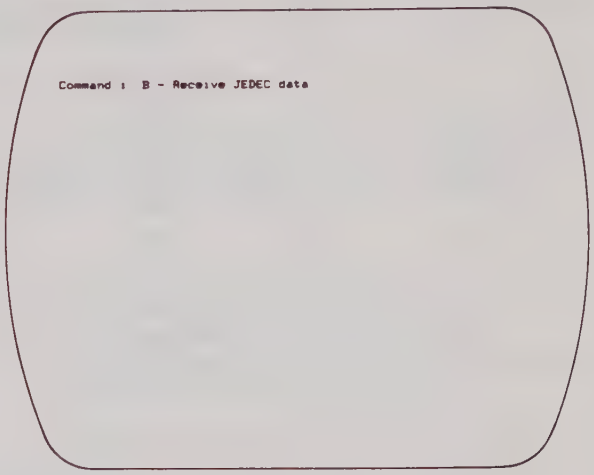
NOTE

□ is the action symbol. XXXX represents the fuse array sum-check.

Terminal Control

To receive JEDEC data from the terminal mode, enter a "B" from the Command mode:

Terminal Displays



3.5.10 EDIT FUSE PATTERN

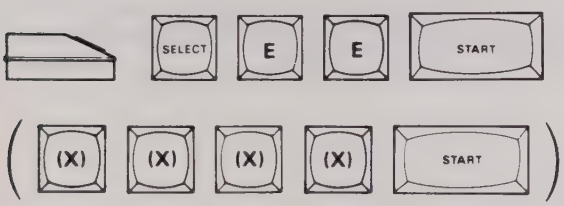
The individual fuses that make up a PAL® fuse map may be edited using the fuse map editor. Fuses may be changed from blown to unblown or vice-versa on a downloaded fuse map, a fuse map generated by assembly of source code, or directly in fuse memory.

Fuses may be edited one at a time from the front panel, or in line editor fashion from a terminal. In the examples that follow, assume that we are editing the Basic Gates fuse map of figure 3-8, representing the logic diagram of figure 3-9.

If "Device Selection Error" appears when you enter the fuse editor, you must specify the family code and pinout code to define the fuse map.

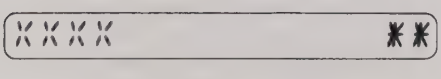
Front Panel Control

Enter the fuse editor with select code EE:



29A Displays

XXXX is decimal number of fuse being edited, ** is binary state of fuse number XXXX (00 or 01).



The desired fuse number for editing from the front panel may be scrolled to by using the START and REVIEW keys, or specified directly by entering the fuse number XXXX as shown above. The data display on the right reflects the current state of the selected fuse:

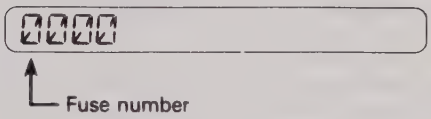
01 = high-resistance, "blown" fuse

00 = low-resistance, fuse intact

Entering a 0 or a 1 while displaying a selected fuse will store that state for the fuse.

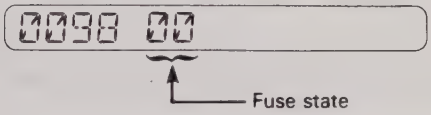
To change fuse number 98 in our Basic Gates example from unblown to blown:

29A Displays



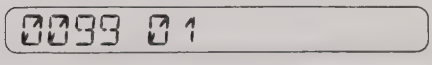
Enter the decimal fuse number, 98.

29A Displays



This display indicates that RAM data for fuse 98 is set for "don't program." To change it to a programmed (blown) state:

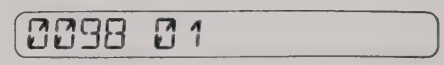
29A Displays



(Fuse number increments automatically.)

To decrement a fuse number:

29A Displays



Terminal Control

Enter an "E" from the terminal Command mode:

Terminal Displays

Command : E - Edit fuse pattern
- COMMANDS -
0 ---------- Display menu
---------- Go to fuse number
CTRL Z ------ Exit fuse editor
- FUSE EDITING -
Space ---------- Move cursor right
BKSP (CTRL H) --- Move cursor left
Return --------- Edit next row
CTRL Z --------- Exit fuse editor
Enter decimal fuse number : 0000

You may now specify a fuse number directly, or enter RETURN to display the first fuse row.

The fuse editor is a fixed-format line editor, with the first column of the displayed line reserved for command characters. A character entered in the first column (normally blank) is interpreted as a command and acted upon immediately; otherwise fuse editing is not processed until a RETURN or CTRL Z is entered (at any point on the line). The command characters recognized in the first column are 0 (zero) and #.

The fuse editor display (see figure 3-14) shows the specified fuse number followed by the next N consecutive fuses, where N is the number of fuses in one row of the selected PAL. Any fuse number may be specified, regardless of row boundaries, and the display will follow this convention. Thus entering RETURN at any time moves the editor to the fuse one row down from the previously specified fuse. Index marks are shown over every tenth fuse in the row displayed, for easy location of fuses beyond the one specified. Also note that the fuse display may be changed from X/- to 0/1 with select code CE or main menu command G (see section 3.5.12).

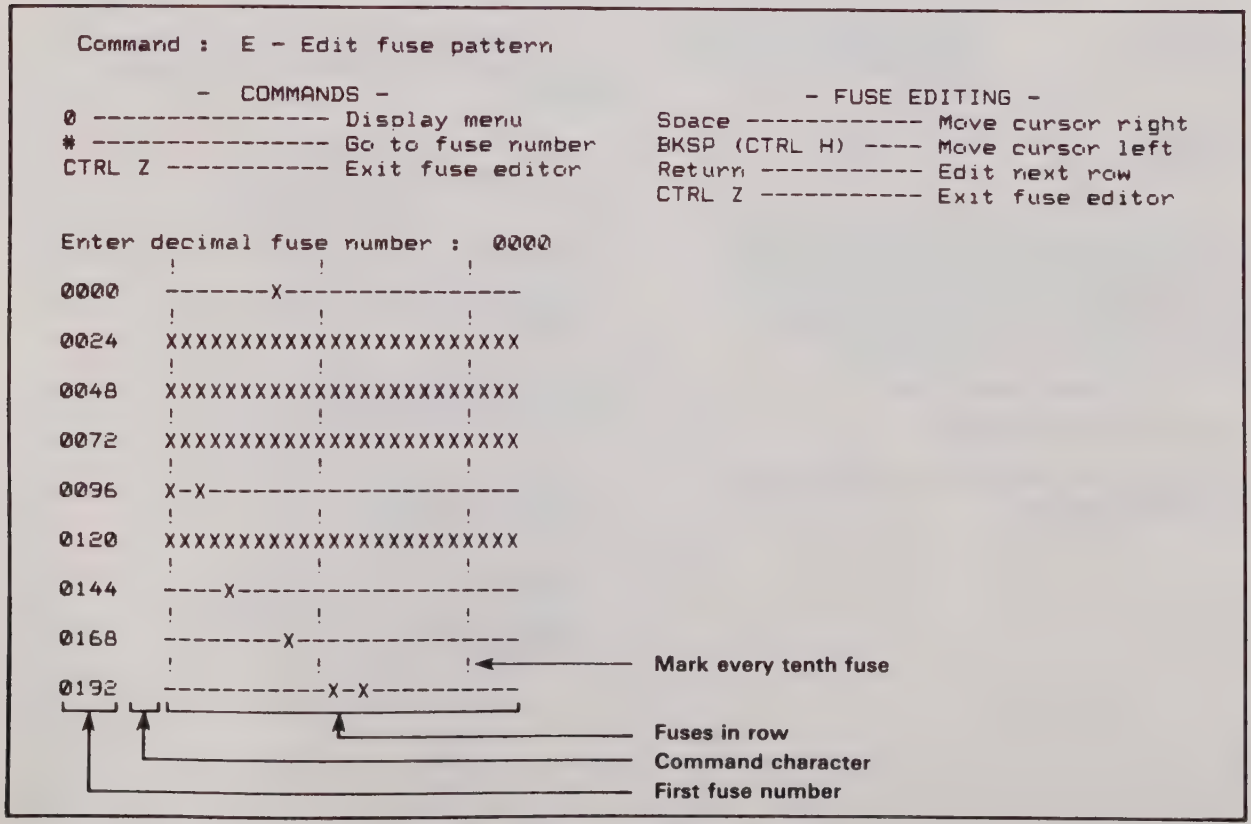


Figure 3-14. Default Fuse Editor Pattern

In operation, the fuse editor copies the selected row to a temporary buffer where all editing changes are made. Then, when a command or RETURN is entered, the editing buffer is examined for legal characters before copying back to the fuse map. You are not allowed to proceed to another row until all characters are legal in the current row. Typing a CTRL Z to exit the fuse editor from an untested edited row will leave the row in its original state.

To edit a fuse row, use the following procedure:

1. Move the cursor back and forth along the displayed row using SPACE and BACKSPACE until it is positioned over the fuse to be changed.
2. Type the desired symbol to enter it into the editing buffer as the fuse state.
3. Enter 0 (zero) or # in the command character position at any time to display the menu or move to a specific fuse number.
4. Type RETURN at any time to move to the next row.
5. Type CTRL Z at any time to exit the fuse editor.

Editing fuse number 98 in our example may be done in two ways. As one method, you can enter the fuse editor and type RETURN until the desired row appears (beginning with fuse 0096), resulting in a display that matches the device data sheet, and then space three times to locate fuse 98. The display in this case will resemble figure 3-14.

Alternatively, fuse number 98 may be directly specified. When this is done, a fuse "row" is displayed which begins with fuse number 0098 and does not match any of the rows in the logic diagram of figure 3-9. Fuse number 98 may now be modified without counting spaces, and subsequent RETURNs will jump to the fuses directly below fuse 98 in the same column (122, 146, 170, etc.). Figure 3-15 shows the display when this method is used.

```
Enter decimal fuse number :  0098

0098   X----------------------XX

0122   XXXXXXXXXXXXXXXXXXXXXX--

0146   --X--------------------

0170   ------X----------------
```

Figure 3-15. Starting Fuse Not on Row Boundary

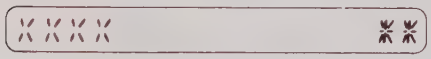
3.5.11 DISPLAY CONFIGURATION NUMBER

This command displays the configuration number of the adapter firmware. Configuration numbers are used as serial numbers for firmware.

Front Panel Control

Enter SELECT EF from the front panel:

29A Displays



Terminal Control

Enter an "F" from the terminal command mode:

Terminal Displays

NOTE

XXXX is the configuration number of the firmware in the adapter plugged into the PLDS.

3.5.12 SELECT ATTRIBUTES

This command allows you to select one of two options for any of seven attributes as shown below. The only options available to the PALASM and H & L Design Adapters are those numbered 0 thru 7:

OPTION	DESCRIPTION
0	Echo (full duplex): PLDS echoes all characters received at the serial port.
1	No echo (half duplex).

NOTE

The default echo mode will depend upon the programmer being used. The 29A and 100A programmers will power up in the "no echo" mode, while the Model 19 will power up in the echo mode.

2	JEDEC full mode: described by the JEDEC standard (JC-42-1-81-62). This is the default state.
3	JEDEC kernel mode: selects the kernel mode (see appendix A for kernel mode definition).
4	Fuse display X/—: displays an unprogrammed fuse as "X" and a programmed fuse as a "-". This is the default state.
5	Fuse display 0/1: displays an unprogrammed fuse as "0" and a programmed fuse as "1".
6	End upload with ETX: PLDS terminates an upload operation (serial data transmission) with an ETX character (ASCII hex 03). This is the default state.
7	End upload with CTRL Z: ends the upload with a CTRL Z.

An underblow condition occurs when the programmer RAM indicates that a particular fuse should be blown and the device in the socket shows the fuse to be unblown. An overblow condition occurs when the programmer RAM indicates that a fuse is unblown yet the part shows it to be blown.

8	Disable underblow/overblow display: disables this attribute.
9	Enable underblow/overblow display: enables this attribute.

Some registered devices need to be initialized to a known state before functional testing. One method is to force the registers to a particular state. This function is referred to as preload. The default state (A) disables the function. The function may be enabled by entering a "B", but if it is not implemented in the P/T adapter, a warning message will be output and the feature once again is disabled.

A	Disable preload option: this is the default state.
B	Enable preload option
C	Two-pass functional verify: performs the normal two-pass functional verify at V_{CC} voltages above and below nominal.
D	One-pass functional verify: speeds up the testing cycle by only doing a one-pass functional verify at the nominal V_{CC} voltage.

Front Panel Control

To access the attributes from the front panel, do the following:

29A Displays

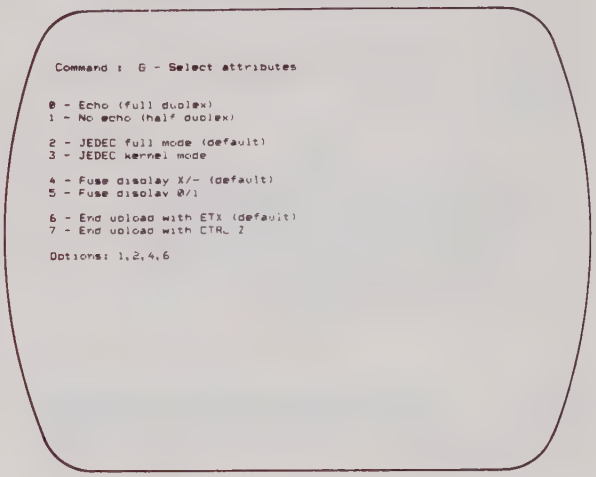
To change any attribute, enter the code number from those given above where the (X) is shown in the following key sequence.

29A Displays

Terminal Control

To access the attributes from the terminal, enter a G from the command mode.

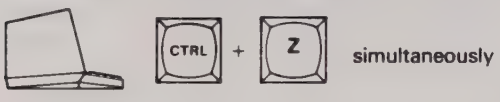
Terminal Displays


Command : G - Select attributes
0 - Echo (full duplex)
1 - No echo (half duplex)
2 - JEDEC full mode (default)
3 - JEDEC kernel mode
4 - Fuse display X/- (default)
5 - Fuse display 0/1
6 - End upload with ETX (default)
7 - End upload with CTRL Z
Options: 1,2,4,6

To change an attribute or attributes from the terminal, space or backspace (CTRL H) to the appropriate attribute(s) and enter the new value. The edit session is terminated by a RETURN if the edited attribute(s) are to be saved or by a CTRL Z if they are not to be saved. If an invalid value is entered the line will be repeated, including the invalid data, waiting for the correct value(s) to be entered.

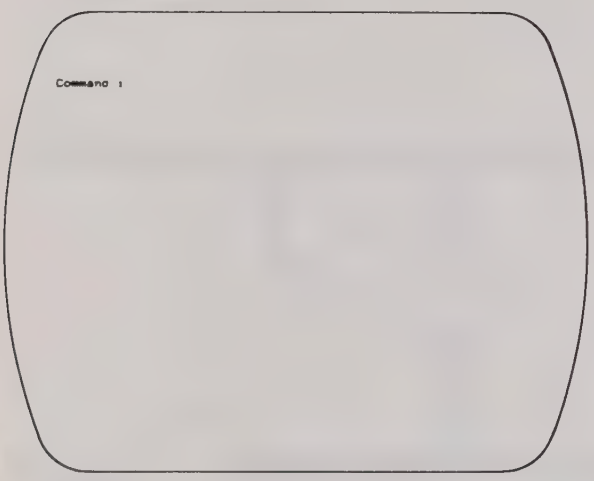
3.5.13 EXIT COMMANDS

During terminal mode, a CTRL Z will exit specific operating modes. When using the design adapters, this function is also used to terminate the change, insert, and edit modes.



simultaneously

Terminal Displays



The ESC key is used to terminate all PLDS operations and return control to the front panel. This must be done before removing an adapter or the LogicPak™.

returns control to the programmer front panel from terminal control.

SECTION 4

CIRCUIT DESCRIPTION

4.1 INTRODUCTION

This section defines the functions of the PALASM Design Adapter's principal components and firmware.

4.2 GENERAL ARCHITECTURE

The adapter interfaces with the LogicPak™. When installed, the adapter's firmware replaces the firmware in the LogicPak™.

4.3 COMPONENT LAYOUT

The block diagram of the adapter's electronics is shown in figure 4-1, and the schematic is at the back of this manual. The design adapter's firmware is resident in two or more EPROMs, which receive address and select inputs from the LogicPak™. The EPROM outputs are buffered by an octal data gate.

One EPROM (U1) resides in the address range 6000 to 7FFF, while the other four EPROMs (U2, U3, U4 and U8) are paged into the address range 8000 to 9FFF. The paging is controlled by the LED select lines (SEL A and SEL B).

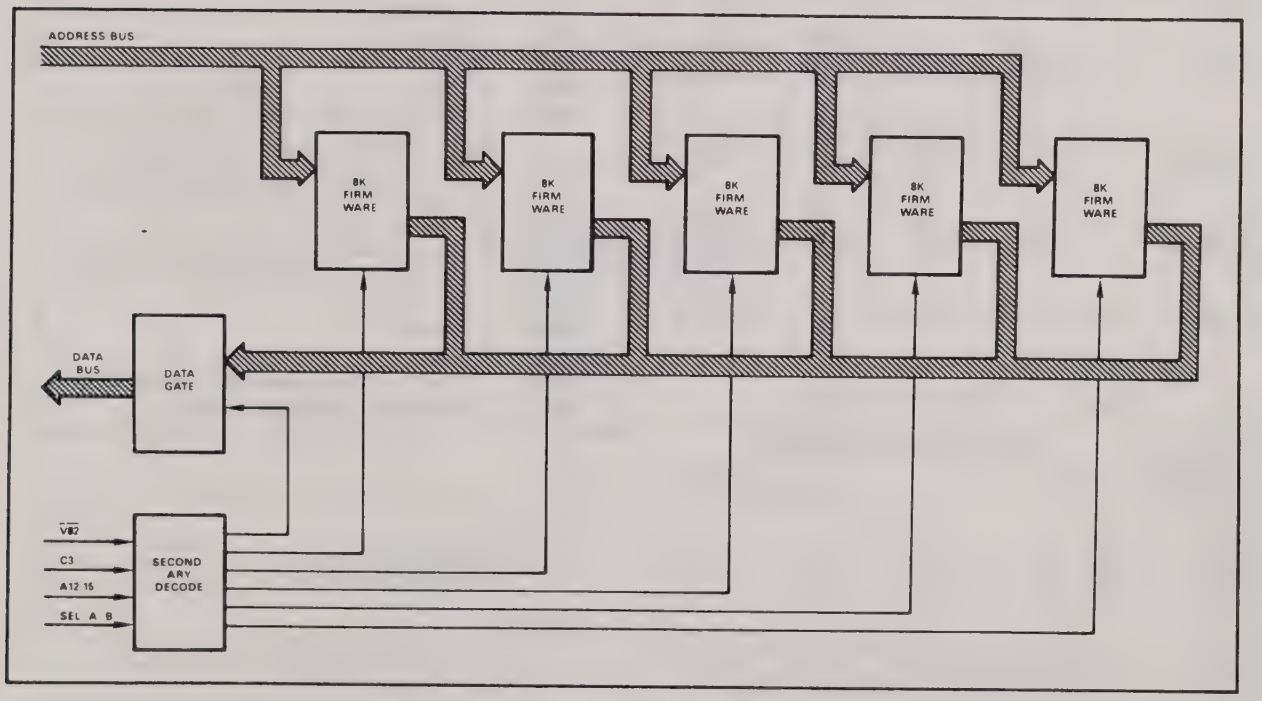


Figure 4-1. Block Diagram, PALASM Design Adapter

4.4 FIRMWARE OPERATION

The data RAM memory map is shown in table 4-1. The fuse data occupies the first 1K bytes of data RAM. Each byte defines 8 fuses, see table 4-2. General test information is contained in 14 bytes starting at address 0400 (see table 4-3). The structured test vectors are stored as ASCII characters starting at address 040F (see figure 4-2).

The PALASM source is stored as ASCII characters (see figure 4-3). The starting location is dependent on the programmer RAM size (see table 4-1). The first byte is always a NUL (ASCII 00). The end of text marker has the eighth bit set, i.e., 80 through FF hex. The carriage return (ASCII OD hex) is the only control character allowed in the source buffer.

Table 4-1. PALASM Design Adapter Memory Map

	PROGRAMMER RAM SIZE			
	4K	8K	16K	64K
Fuse	0000 -- 03FF	0000 -- 03FF	0000 -- 03FF	0000 -- 03FF
Test	0400 -- 040E	0400 -- 040E	0400 -- 040E	0400 -- 040E
Vector	040F -- 07FF	040F -- 0FFF	040F -- 1FFF	040F -- 1FFF
Source	0800 -- 0FFF	1000 -- 1FFF	2000 -- 3FFF	2000 -- 3FFF

Table 4-2. Fuse RAM Memory Map

RAM ADDRESS	DECIMAL FUSE NUMBERS MSB							LSB
0000	7	6	4	4	3	2	1	0
0001	15	14	13	12	11	10	9	8
0002	23	22	21	20	19	18	17	16
0003	31	30	29	28	27	26	25	24
--	--	--	--	--	--	--	--	--
03FE	8183	8182	8181	8180	8179	8178	8177	8176
03FF	8191	8190	8189	8188	8187	8186	8185	8184

Table 4-3. Test Information Memory Map

RAM ADDRESS	DESCRIPTION
0400	Test cycle count
0401	Starting vector for Logic Fingerprint™ test (4 bytes)
0402	Same
0403	Same
0404	Same
0405	Structured vector test: vector that failed Transmit JEDEC data: CheckSum MSB
0406	Structured vector test: pin that failed Transmit JEDEC data: CheckSum LSB
0407	Security fuse state
0408	Receive JEDEC data: field that failed
0409	Resulting Logic Fingerprint™ (4 bytes)
040A	Same
040B	Same
040C	Same
040D	Flag to indicate structured vectors present
040E	Number of structured vectors

```
ADDRESS
   040F  XXXXXXXXXNXXXXXXXXN          20 pin vector #1
   0423  XXXXXXXXXNXXXXXXXXN          20 pin vector #2
   0437  XXXXXXXXXNXXXXXXXXN          20 pin vector #3

ADDRESS
   040F  XXXXXXXXXXXNXXXXXXXXXXXN     24 pin vector #1
   0427  XXXXXXXXXXXNXXXXXXXXXXXN     24 pin vector #2
   043F  XXXXXXXXXXXNXXXXXXXXXXXN     24 pin vector #3
```

Figure 4-2. Test Vector Memory Map

```
                        HEX                                    ASCII

00 50 41 4C 31 32 48 36 20 20 20 20 20 20 50 41   .PAL12H6      PA
4C 20 44 45 53 49 47 4E 20 53 50 45 43 49 46 49   L DESIGN SPECIFI
43 41 54 49 4F 4E 20 0D 50 30 31 0D 42 41 53 49   CATION .P01.BASI
43 20 47 41 54 45 53 0D 44 41 54 41 20 49 2F 4F   C GATES.DATA I/O
0D 43 20 44 20 46 20 47 20 4D 20 4E 20 50 20 51   .C D F G M N P Q
20 49 20 47 4E 44 20 4A 20 4B 20 4C 20 52 20 4F    I GND J K L R O
20 48 20 45 20 42 20 41 20 56 43 43 0D 0D 42 20    H E B A VCC..B
3D 20 2F 41 20 20 20 20 20 3B 49 4E 56 45 52 54   = /A     ;INVERT
45 52 0D FF FF 00 00 00 00 00 00 00 00 00 00 00   ER.............
```

Figure 4-3. Text Source Buffer

APPENDIX A

STANDARD DATA TRANSFER FORMAT BETWEEN DATA PREPARATION SYSTEM AND PROGRAMMABLE LOGIC DEVICE PROGRAMMER

This document defines a format for the transfer of information between a data preparation or storage system and a device programmer. This format provides for, but is not limited to, the transfer of fuse, test, identification, and comment information in an ASCII representation. This format defines the "intermediate code" between device programmers and data preparation storage systems.

NOTE

This is Data I/O's implementation of the JEDEC (Joint Electron Device Engineering Council) standard (JC-42.1-81-62).

Copyright 1983

Data I/O Corporation
10525 Willows Road N.E./C-46
Redmond, WA 98052
(206) 881-6444

Version 1.1 May 5, 1983

SECTION A.1

INTRODUCTION

The Backus-Naur Form (BNF) is used in this document to define the format syntax.

A.1.1 BNF RULES

"::=" denotes "is defined as".

Characters enclosed by single quotes are literals.

Parentheses enclose indentifiers.

Square brackets enclose optional items.

Braces (curly brackets) enclose a repeated item. The item may appear zero or more times.

A vertical bar separates alternative items.

A repeat count is given by a ":n" suffix. For example, a six digit number would be defined as (number) ::= (digit) :6.

Example

The English description of a person's name may be as follows:

> The full name consists of an optional title followed by a first name, a middle name, and a last name. The person may not have a middle name or may have several middle names. The titles consist of : Mr., Mrs., Ms., Miss, and Dr.

The BNF definition would be:

(full name) ::= [(title)] (f. name) { (m. name)}
(l. name)

(title) ::= 'Mr.' | 'Mrs.' | 'Ms.' | 'Miss' | 'Dr.'

A.1.2 BNF DEFINITIONS

(digit) ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

(hex-digit) ::= (digit) | 'A' | 'B' | 'C' | 'D' | 'E' | 'F'

(binary-digit) ::= '0' | '1'

(number) ::= (digit) {(digit)}

(del) ::= (space) | (carriage return)

(delimiter) ::= (del) {(del)}

(printable character) ::= (ASCII 20 hex ... 7E hex)

(control character)::= (ASCII 00 hex ... 1F hex)
| (ASCII 7F hex)

(STX) ::= (ASCII 02 hex)
(ETX) ::= (ASCII 03 hex)
(carriage return) ::= (ASCII 0D hex)
(line feed) ::= (ASCII 0A hex)
(space) ::= (ASCII 20 hex) | ' '

(valid character) ::= (printable character)
| (carriage return) | (line feed)

(field character) ::= (ASCII 20 hex ... 2A hex)
| (ASCII 2C hex ... 7E hex)
| (carriage return) | (line feed)

TRANSMISSION PROTOCOL

The transmission consists of a start-of-text (STX) character, the various fields, an end-of-text (ETX) character, and a transmission check-sum. The character set consists of the printable ASCII characters and four control characters (STX, ETX, CR, LF). The other control characters should not be used because they can produce undesirable side-effects in the receiving equipment.

BNF Syntax

(format) ::= (STX) (design spec) {(field)} (ETX)
(xmit check-sum)

A.2.1 DESIGN SPEC

The design specification is the first field in the format and doesn't have an identifier. An asterisk terminates the field. This field must be included. The design specification should consist of:

1. User's name and company
2. Date, part number, and revision
3. Manufacturer's device number
4. Other information

BNF Syntax

(design spec) ::= {(field character)} '*'

A.2.2 TRANSMISSION CHECK-SUM

The transmission check-sum is the 16-bit sum (i.e., modulo 65,535) of all ASCII characters transmitted between and including the STX and ETX. The parity bit is excluded in the calculation.

BNF Syntax

(xmit check-sum) ::= (hex-digit):4

A.2.3 FIELDS

Each field starts with a single character identifier. Multiple character identifiers could be used to create sub-fields (i.e., "A1", "A$", or "AB3"). The field is terminated with an asterisk and therefore asterisks cannot be imbedded within the field. While not required, carriage returns and line feeds should be used to improve the readability of the format.

BNF Syntax

(field) ::= [(Delimiter)] (field identifier)
{(field character)} '*'

(field identifier) ::= 'C' | 'D' | 'F' | 'G' | 'L' | 'M' | 'P' | 'Q'
| 'R' | 'S' | 'T' | 'V'

(reserved identifier) ::= 'A' | 'B' | 'E' | 'H' | 'I' | 'J' | 'K'
| 'N' | 'O' | 'U' | 'W' | 'X' | 'Y' | 'Z'

Identifiers

A - *	N - *
B - *	O - *
C - Check sum	P - Pin sequence
D - Device type	Q - Value
E - *	R - Resulting vector
F - Default fuse state	S - Starting vector
G - Security fuse	T - Test cycles
H - *	U - *
I - *	V - Test vector
J - *	W - *
K - *	X - *
L - Fuse list	Y - *
M - Option	Z - *

* Reserved for future use

SAMPLE TRANSMISSION:

```
(STX)
Acme Logic Design          Joan Engineer          Feb. 29 1983
Widget Decode 756-AB-3456 Rev C  Device Mullard  12AX7*
QP20* QF384* G1*
L0000 1111111011 1111111111 1111000000 0000000000
      0000000000 0000000000 0000000000 0000000000
      0000000000 0000000101 1111111111 1111111111
      0000000000 0000000000 0000111101 1111111111
      1111111111 1111110111 1111111111 1111111111*

L0200 1110101111 1111110000 0000000000 0000000000
      1111111111 1111011011 1111111111 1111111110
      0111111111 1111111111 1111111110 1111111111
      1111111111 1111101111 1111111111 1111101111
      0000000000 0000000000 0000*       C1BB9*

V0001  XXXXXXXXXNXXXXXXXH0N* V0002  XXXXXXXXXNXXXXXXXL1N*
V0003  00XXXXXXXNXXXXXXLXXN* V0004  01XXXXXXXNXXXXXXLXXN*
V0005  10XXXXXXXNXXXXXXLXXN* V0006  11XXXXXXXNXXXXXXHXXN*
V0007  XX00XXXXXNXXXXXLXXXN* V0008  XX01XXXXXNXXXXXHXXXN*
V0009  XX10XXXXXNXXXXXHXXXN* V0010  XX11XXXXXNXXXXXHXXXN*

T02*   S10101010101010101010*   R1ACB678F*
(ETX)  32EB
```

FORMAT FIELD DEFINITIONS

A.3.1 DEVICE TO BE PROGRAMMED

The device field defines the specific device being programmed.

BNF Syntax

(device) ::= 'D' (family-pinout code) '*'

Example

D9501*

A.3.2 FUSE INFORMATION

Each device fuse link is assigned a decimal number. The fuse may have two possible states: a zero to specify a low resistance link and a one to specify a high resistance link.

The fuse information is presented in three fields: the default state (F), the fuse list (L), and the fuse check-sum (C). The default state and fuse check-sum fields are optional.

The "F" field is used to define the states of links not explicitly defined in the "L" fields. If the default state is not used, all device links must be defined in the "L" fields.

The "L" field can be any convenient length, and any number of "L" fields can be used. If the state of a link is specified 2 or more times, the last state replaces the preceding entries. The number following the "L" indicates the starting fuse number for the data string.

The link information check-sum is computed by performing a 16 bit addition of 8 bit words constructed from the specified state of each link in the device. The unused bits in the last word are set to zero. The 8 bit words are defined as shown in the following diagram.

word 00	msb							lsb
Link No.	7	6	5	4	3	2	1	0
word 01	msb							lsb
Link No.	15	14	13	12	11	10	9	8
...								
word 62	msb							lsb
Link No.	-	-	-	-	499	498	497	496

BNF Syntax

(fuse information) ::= [(default state)] (fuse list) {(fuse list)} [(fuse check-sum)]

(default state) ::= 'F' (binary-digit) '*'

(fuse list) ::= 'L' (number) (delimiter) {(binary-digit) [(delimiter)]} '*'

(fuse check-sum) ::= 'C' (hex-digit):4 '*'

Example

F0*L0 01010101* L0008 01010111*L1000 0101*CF3BA*

Example

L0200 01101010101010101011010111010110100010010010010*

A.3.3 STRUCTURED FUNCTIONAL TEST INFORMATION

This field specifies one of several test conditions for each pin of a device. A test vector is defined as N test conditions where N is the number of pins on the device.

Test Conditions

- 0 - Drive input low
- 1 - Drive input high
- 2-9 - Drive input to supervoltage #2-9
- C - Drive input low, high, low
- K - Drive input high, low, high
- N - Power pins and outputs not tested
- L - Test output low
- H - Test output high
- Z - Test input or output for high impedance
- F - Float input or output

The C and K driving signals are presented after the other inputs are stable. The L, H, and Z tests are performed after all inputs have stabilized including C and K.

The test vectors will be applied to the device under test in numerical order. If any vectors are specified 2 or more times the data in the last vector replaces the previous data. There is no default test vector.

The optional "P" field is used to specify the correspondence of the test conditions to the device pin numbers. If the "P" field is not used, the first condition will be applied to pin 1, the second to pin 2, and so on through pin N.

BNF Syntax

(function test) ::= [(pin list)] (test vector) {(test vector)}

(pin list) ::= 'P' (pin number):N '*'

(pin number) ::= (delimiter) (number)

N ::= number of pins on device

(test vector) ::= 'V' (number) (delimiter) (test condition):N '*'

(test condition) ::= (digit) | 'C' | 'F' | 'H' | 'K' | 'L' | 'N' | 'Z'

(reserved condition) ::= 'A' | 'B' | 'D' | 'E' | 'G' | 'I' | 'J' | 'M' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z'

Example

```
P 11  12  13  14  15  16  17  18  19  20
10   9   8   7   6   5   4   3   2   1*

V0001 C01010101NHLLLHHLHLN*
V0002 C01011111NHLLHLLLHLN*
V0003 C10010111NZZZZZZZZZZN*

V4 C01010100NFLHHLFFLLN*
```

A.3.4 OPTIONAL INFORMATION

Additional option fields may be defined using the letters G, S, R, M, Q, and T. Each field must begin with one of the above letters and be terminated with an asterisk (*). No other restrictions are applied. Therefore, multiple letters could be used to create any number of option fields.

BNF Syntax

(option field) ::= (option ident.) {(field character)} '*'

(option ident.) ::= 'G' | 'S' | 'R' | 'M' | 'Q' | 'T'

Example

```
Q*
MFG Acme Semiconductor*
M:1234*
```

Data I/O uses five optional fields; three for the Logic Fingerprint™ test, a security fuse field, and a value field.

LOGIC FINGERPRINT™

The 'S' field defines the starting vector for the Logic Fingerprint™ test. The possible states are 0 (TTL low) and 1 (TTL high). The 'R' field contains the resulting vector or test-sum. The 'T' field denotes the number of test cycles to be run.

BNF Syntax

(starting vector) ::= 'S' (test condition):N '*'

(resulting vector) ::= 'R' (hex-digit):8 '*'

(test cycles) ::= 'T' (number) '*'

N ::= number of pins on device

Example

```
S0100010000111000111101l0*
R5BCD34A7*
T01*
```

SECURITY FUSE

The security fuses of certain logic devices may be enabled for programming by sending a "1" in the "G" field.

BNF Syntax

(security fuse) ::= 'G' (binary-digit) '*'

Example

G1*

VALUES

The "Q" field is used to express values or limits required by the receiving device. Two subfields have been defined, the "P" subfield for number of pins on the device and the "F" subfield for the number of fuses. These two values will enable the receiving device process the other fields without knowing the device manufacturer or family/pinout code.

BNF Syntax

(number of pins) ::= 'QP' (number) '*'

(fuse limit) ::= 'QF' (number) '*'

Example

QP24* QF1024*

SECTION A.4

OTHER RULES

A.4.1 TRANSPORTABILITY

All receiving machines should have a "kernel" mode to ignore all optional fields so that the actual programming data will be transportable. For example, as allowed in the format, optional fields can be sent to specify additional checksums. A receiving machine in the "kernel" mode could ignore this information yet receive the link information required to program the device.

If the optional "F" field is used to avoid transmitting link data, transportability could be lost. Therefore, whenever practical, data should be transmitted for all links of the device.

Some computer operating systems add control characters after each line making it very difficult to compute the transmission check-sum. To disable the transmission check-sum, receiving equipment should accept "0000" as a valid check-sum.

BNF Syntax

(kernel) ::= (STX) (design spec) (min. fuse information) (ETX) (xmit check-sum)

(design spec) ::= {(field character)} '*'

(min. fuse information) ::= (fuse list) {(fuse list)}

Example:

```
(STX)
Acme Logic Design          Jane Engineer          Feb. 29 1983
Widget Decode 756-AB-3456 Rev C Device Mullard 12AX7*

L0000 1111111011 1111111111 1111000000 0000000000
      0000000000 0000000000 0000000000 0000000000
      0000000000 0000000101 1111111111 1111111111
      0000000000 0000000000 0000111101 1111111111
      1111111111 1111110111 1111111111 1111111111*

L0200 1110101111 1111110000 0000000000 0000000000
      1111111111 1111011011 1111111111 1111111110
      0111111111 1111111111 1111111110 1111111111
      1111111111 1111101111 1111111111 1111101111
      0000000000 0000000000 0000*

(ETX)00000
```

A.4.2 RESTRICTIONS AND VARIATIONS

VARIATIONS FROM PRESENT JEDEC STANDARD

a) The delimiter after the link number or the vector number may be any combination of carriage returns, line feeds, and/or spaces.

b) The check-sum and sum-check are renamed fuse check-sum and xmit check-sum respectively.

VARIATIONS FOR 303A-VO1 LOGICPAK™

a) The link number ("L" field) and the vector number ("V" field) must be 4-digit numbers.

b) The delimiter after the link number and the vector number must include a space.

c) Spaces are not allowed between the terminating asterisk of one field and the identifier of the next field.

d) The security fuse ("G" field) requires a space after the binary digit.

e) The kernel mode is not implemented.

f) The "D" field must have a valid family and pinout code.

(device) ::= 'D' (hex-digit):4 '*'

APPENDIX B

DEVICE INFORMATION ERROR CODES

APPENDIX B

This appendix contains a list of PALASM family and pinout codes (table B-1), a list of error codes (table B-2), and a package of logic diagrams (figures B-1 through B-37). The use of these diagrams is explained below.

The logic diagrams describe the architecture of all the PALs that PALASM supports. Each fuse is represented by the intersection of an input line (vertical) and a "product term" line (horizontal). A blank diagram represents a PAL with all fuses blown (programmed).

Useful logic equations may be defined by showing inputs connected to output AND gates with an X, representing a fuse left intact (unprogrammed). An example is shown in figure 1-5, which represents the logic equations of figure 1-4.

The equivalence between the logic diagrams, the PALASM fuse display, JEDEC fuse data, and the physical state of the fuse is shown below.

Each fuse in a PAL is numbered. You may find the number of a given fuse on the logic diagram by taking the following steps:

1. Locate the fuse intersection on the logic diagram.
2. Follow the product-term line to its left end and read the first fuse number for the row.
3. Follow the input line from the fuse intersection to either the top or bottom end and read the increment number.
4. Compute the fuse number by adding the first fuse number and the increment number.

Logic Diagram	Fuse Display	JEDEC Fuse Data	Fuse State
	X or Ø (selectable)	Ø	Intact (unprogrammed)
	— or 1 (selectable)	1	Blown (programmed)
	...XXXXX... or ...ØØØØØ... (selectable)	...ØØØØØ...	Entire row intact (unprogrammed)

Table B-1. PALASM Family and Pinout Codes

INDUSTRY PART NUMBER	FAMILY AND PINOUT CODE		SOFTWARE VERSION
20-pin PALs			
10H8	E1	18	V02
10L8	E1	13	V02
12H6	E1	19	V02
12L6	E1	14	V02
14H4	E1	20	V02
14L4	E1	15	V02
16A4	E1	24	V02
16C1	E1	21	V02
16H2	E1	22	V02
16H8	E1	25	V02
16HD8	E1	25	V02
16L2	E1	16	V02
16L8	E1	17	V02
16LD8	E1	17	V02
16P8	E1	29	V02
16R4	E1	24	V02
16R6	E1	24	V02
16R8	E1	24	V02
16X4	E1	24	V02
24-pin PALs			
12H10	E1	07	V02
12L10	E1	01	V02
14H8	E1	08	V02
14L8	E1	02	V02
16H6	E1	09	V02
16L6	E1	03	V02
18H4	E1	10	V02
18L4	E1	04	V02
20C1	E1	12	V02
20H2	E1	11	V02
20L2	E1	05	V02
20L8	E1	26	V02
20L10	E1	06	V02
20R4	E1	24	V02
20R6	E1	27	V02
20R8	E1	27	V02
20X4	E1	23	V02
20X8	E1	23	V02
20X10	E1	23	V02

Table B-2. PLDS Error Codes

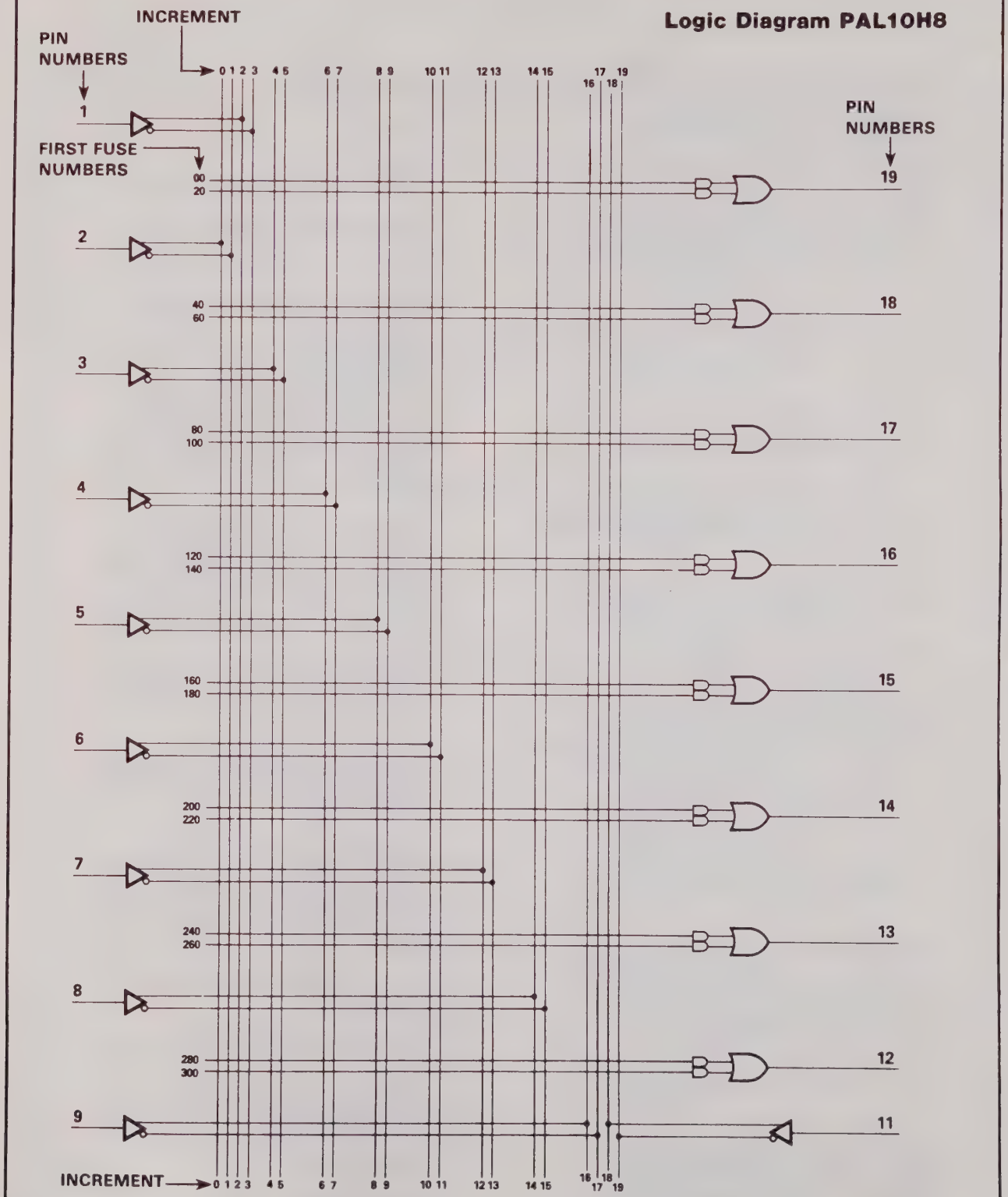
ERROR CODE	DESCRIPTION	ACTION
21*	Illegal Bit Error	The programmer is asked to leave intact a fuse which is already blown. Examine programmer RAM and the device's data.
22*	Programming Error	An attempt to blow a fuse was made and failed. Try the programming sequence again. If the second attempt also fails, try another device. If both devices produce this error, check the calibration of the LogicPak™. If calibrated correctly contact your local Data I/O office.
25*	No Socket Adapter	Insert appropriate socket adapter.
30	No (or Invalid) Device Selected	Enter valid device family and pinout codes (refer to table B-1).
31*	Overcurrent	Hardware error in LogicPak™ or shorted device. Substitute a known-good device or consult the troubleshooting section.
32*	Backward Device/V_{CC} Overcurrent	(1) Device plugged in backward; turn it around. (2) V_{CC} overcurrent, probably caused by a faulty device.
34	Invalid Device Selected	Incorrect family pinout code entered. Enter proper family pinout code. This error occurs in computer remote control only.
35	Source Equation Translation Error	Check equation buffer by connecting terminal to examine the equation buffer. This error code lets the operator know that an error exists in the source equations when the programmer is not controlled by a terminal.
36	Begin RAM Pointer Not = 0000	Refer to programmer manual to reset the begin RAM pointer to zero. This error usually occurs when changing from one programming pak to another.
37	Invalid Device-Related Operation	Verify, program, or other illegal operation was attempted, with a design adapter installed.
38*	Calibration Step Error	(1) Indicates you have selected an incorrect calibration step, or (2) a program operation is attempted prior to exiting calibration—exit the calibration mode (refer to the programmer manual).
63	RAM Write Error	System RAM failure. Refer to programmer manual or contact Data I/O service representative.
65	Firmware Sum-Check Error	Contact Data I/O service representative. This indicates that the EPROM firmware in the LogicPak™ or adapter may have changed since the unit was shipped. Do not continue operation until the situation is corrected.

*These errors do not apply to design adapters.

Table B-2. PLDS Error Codes (Cont.)

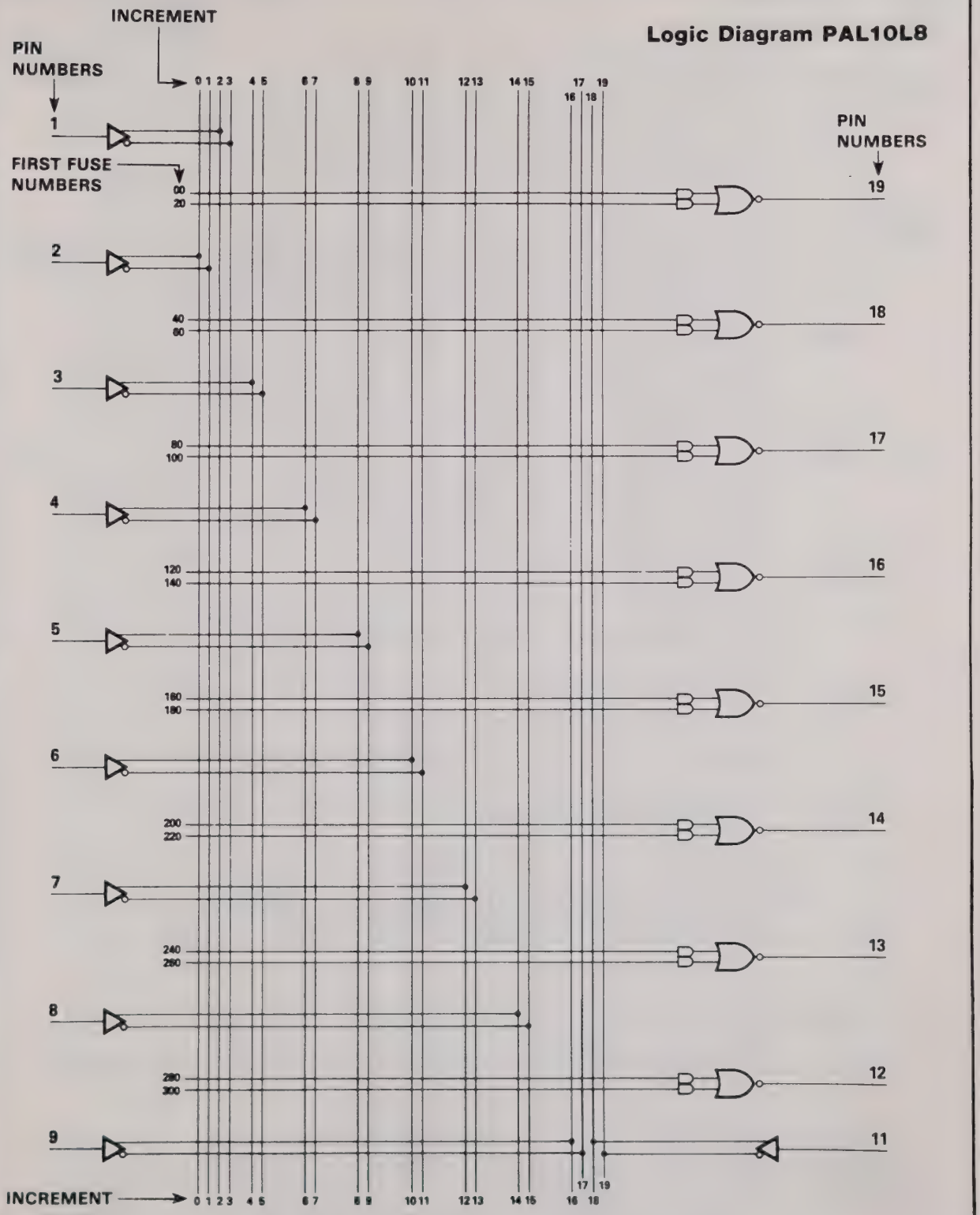
ERROR CODE	DESCRIPTION	ACTION
70*	DAC Error, V_{CC}	See section 4.4 of LogicPak™ manual
71*	DAC Error, Bit Switch Number 1	See section 4.4 of LogicPak™ manual.
72*	DAC Error, Bit Switch Number 2	See section 4.4 of LogicPak™ manual.
73*	DAC Error, CE	See section 4.4 of LogicPak™ manual.
74*	Logic Fingerprint™ Test Verify Error	Indicates one of the following Logic Fingerprint™ errors: (1) Device passed fuse verify but failed Logic Fingerprint™ Test—defective device. (2) Operator has entered wrong test-sum. (3) Device cannot be tested with Logic Fingerprint™ (refer to P/T adapter manual for the limitations of the Logic Fingerprint™ test).
75	Structured Test Verify Error	The device passed fuse verify but failed structured test—defective device. Check structured test vectors and make sure they are correct. If not, reenter the correct vectors. The vector could be invalid, or the operator may have miskeyed a valid vector.
76	Self-Test Error	Indicates failure in the LogicPak™. Consult section 4.4 of your LogicPak™ manual (troubleshooting) or contact your Data I/O service representative.
77	Security Fuse Programming Error	(1) Indicates that the security fuse option cannot be programmed in the installed device, or (2) there is no security fuse option available for this type of device.
78*	No Fuse Verify Set	Indicates you have tried to program the device with the verify-option mode set for 2. The verify option will not allow this. When this error code displays, select E6 and enter Ø or 1, and then you will be allowed 1 program.
79*	Preload Not Implemented	The preload algorithm is not implemented for this device.
81	Parity Error	A parity error occurred while receiving serial data.
82	Checksum Error	Indicates an incorrect transmission data from a peripheral to the serial port, including fuse data, CRs, STX, etc.
84	Invalid Data	See section 3.5.9.
91	Fuse Address Error	See section 3.5.9.

*These errors do not apply to design adapters.



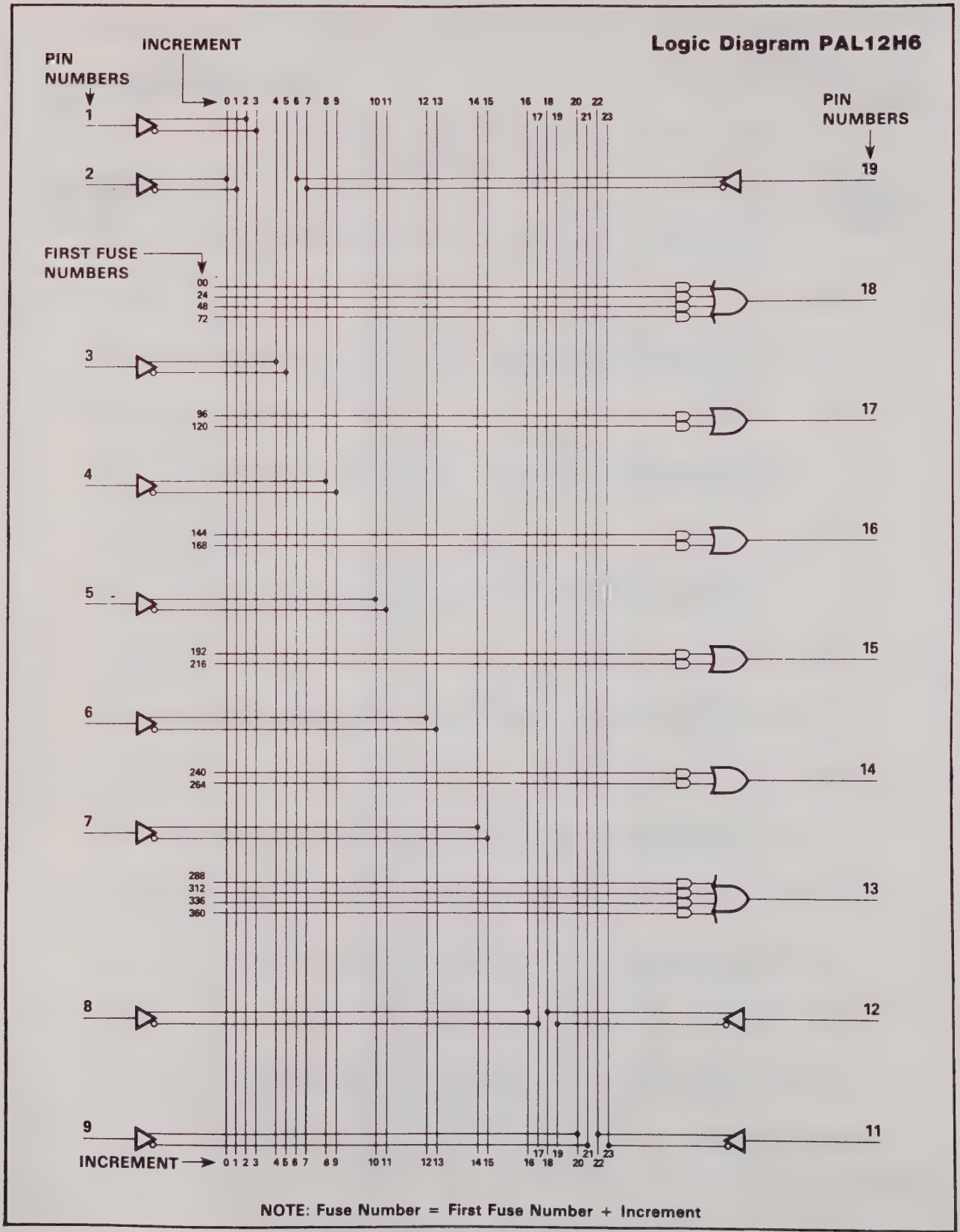
NOTE: Fuse Number = First Fuse Number + Increment

Figure B-1. Logic Diagram PAL10H8



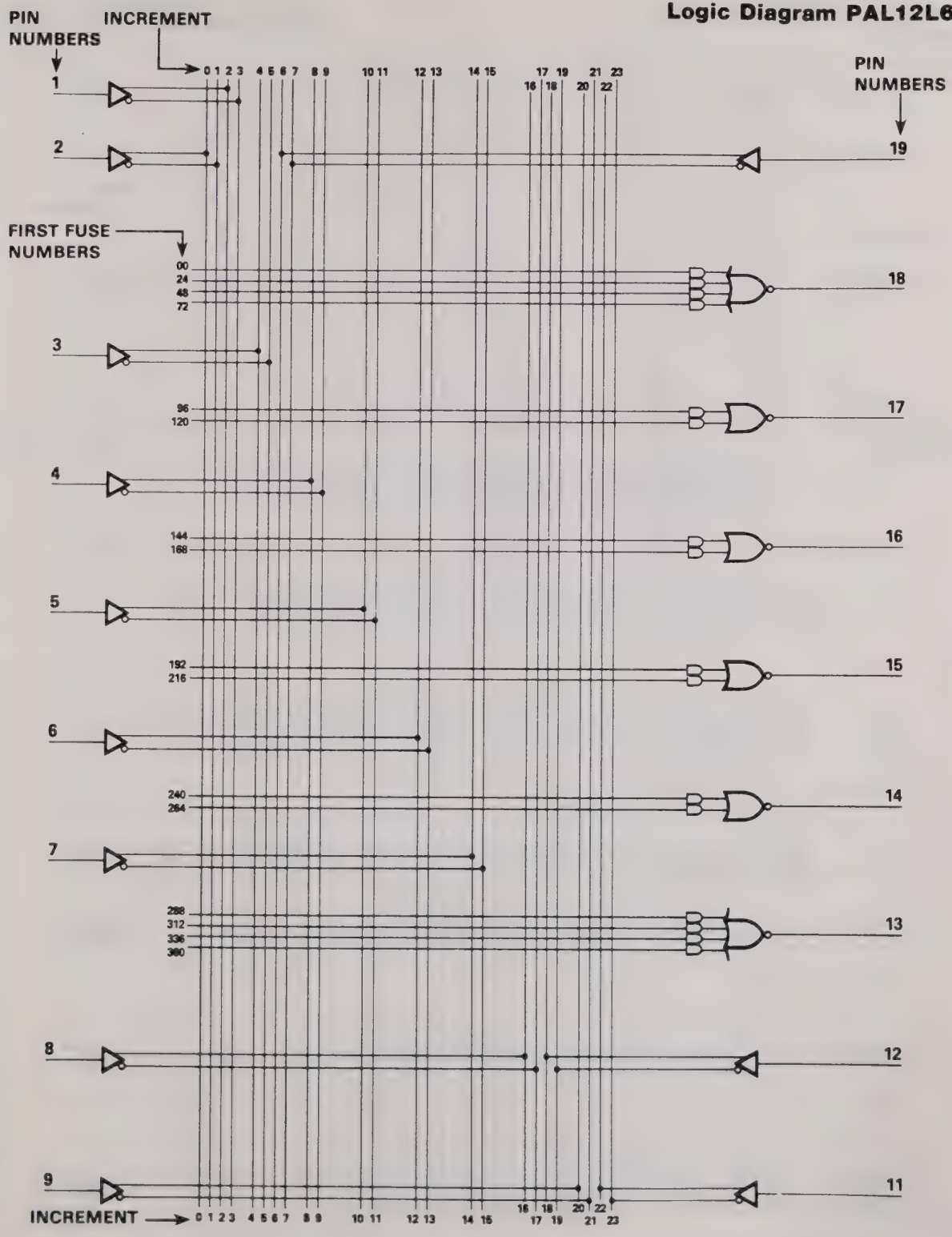
NOTE: Fuse Number = First Fuse Number + Increment

Figure B-2. Logic Diagram PAL10L8



NOTE: Fuse Number = First Fuse Number + Increment

Figure B-3. Logic Diagram PAL12H6



NOTE: Fuse Number = First Fuse Number + Increment

Figure B-4. Logic Diagram PAL12L6

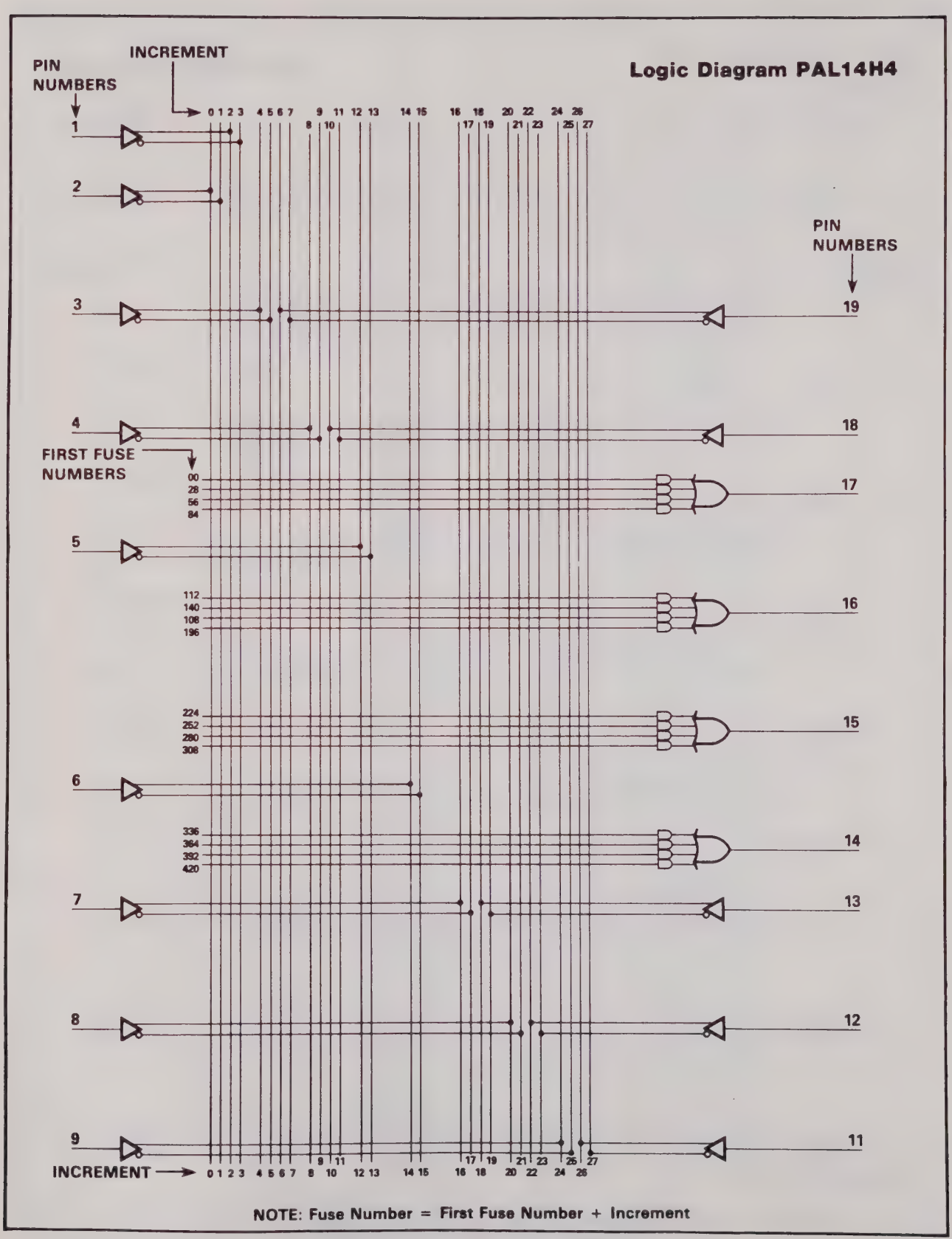
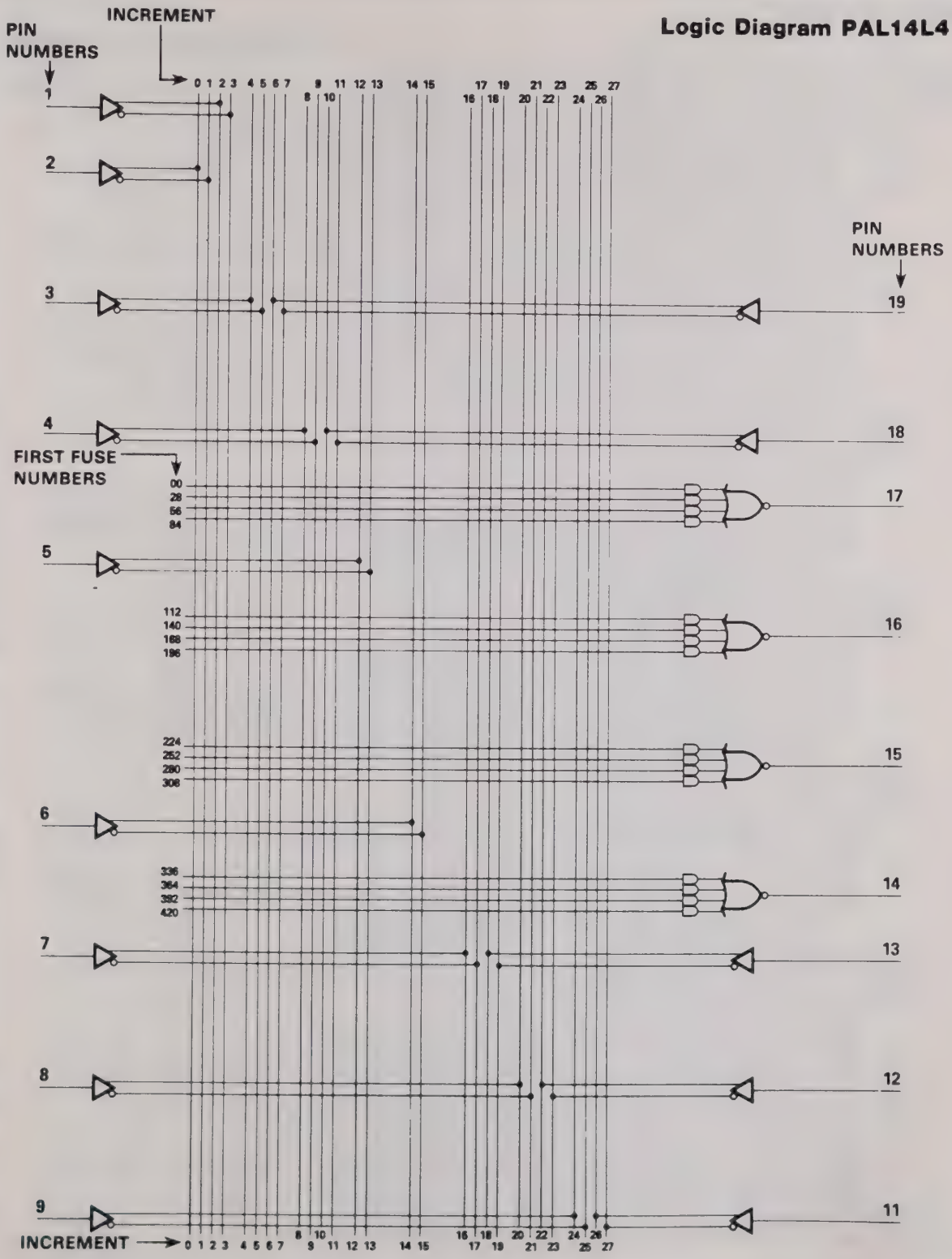


Figure B-5. Logic Diagram PAL14H4



NOTE: Fuse Number = First Fuse Number + Increment

Figure B-6. Logic Diagram PAL14L4

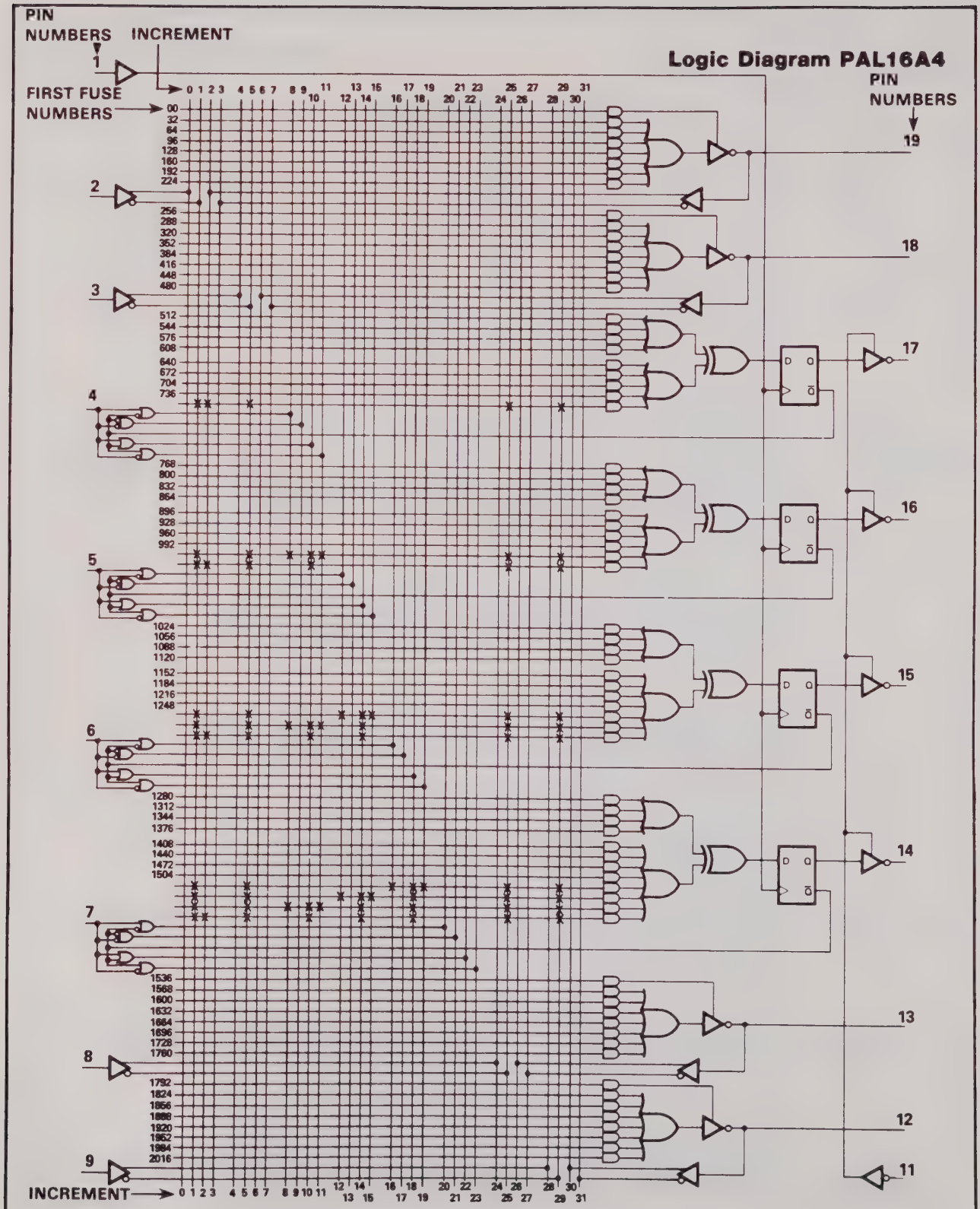


Figure B-7. Logic Diagram PAL16A4

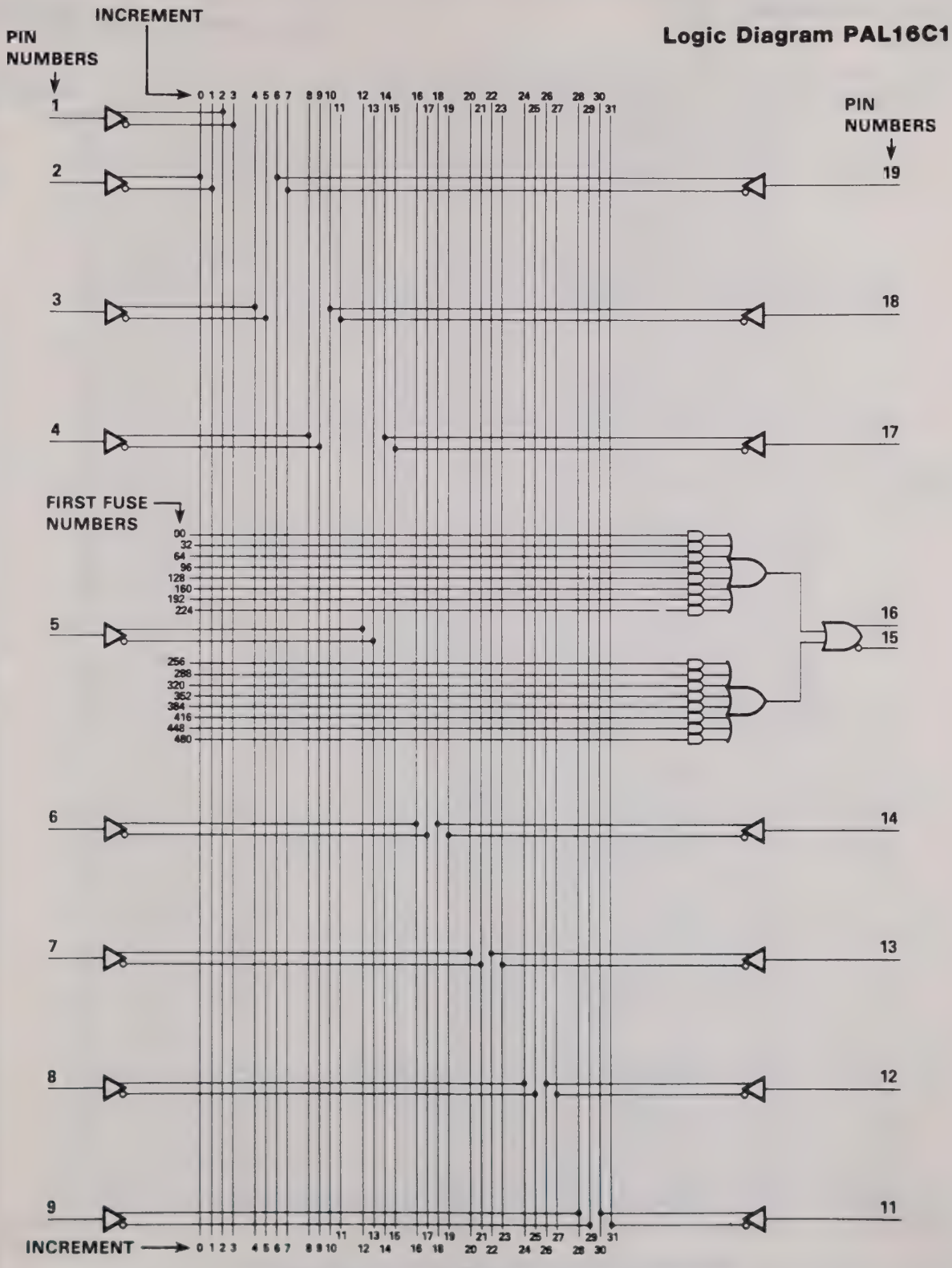


Figure B-8. Logic Diagram PAL16C1

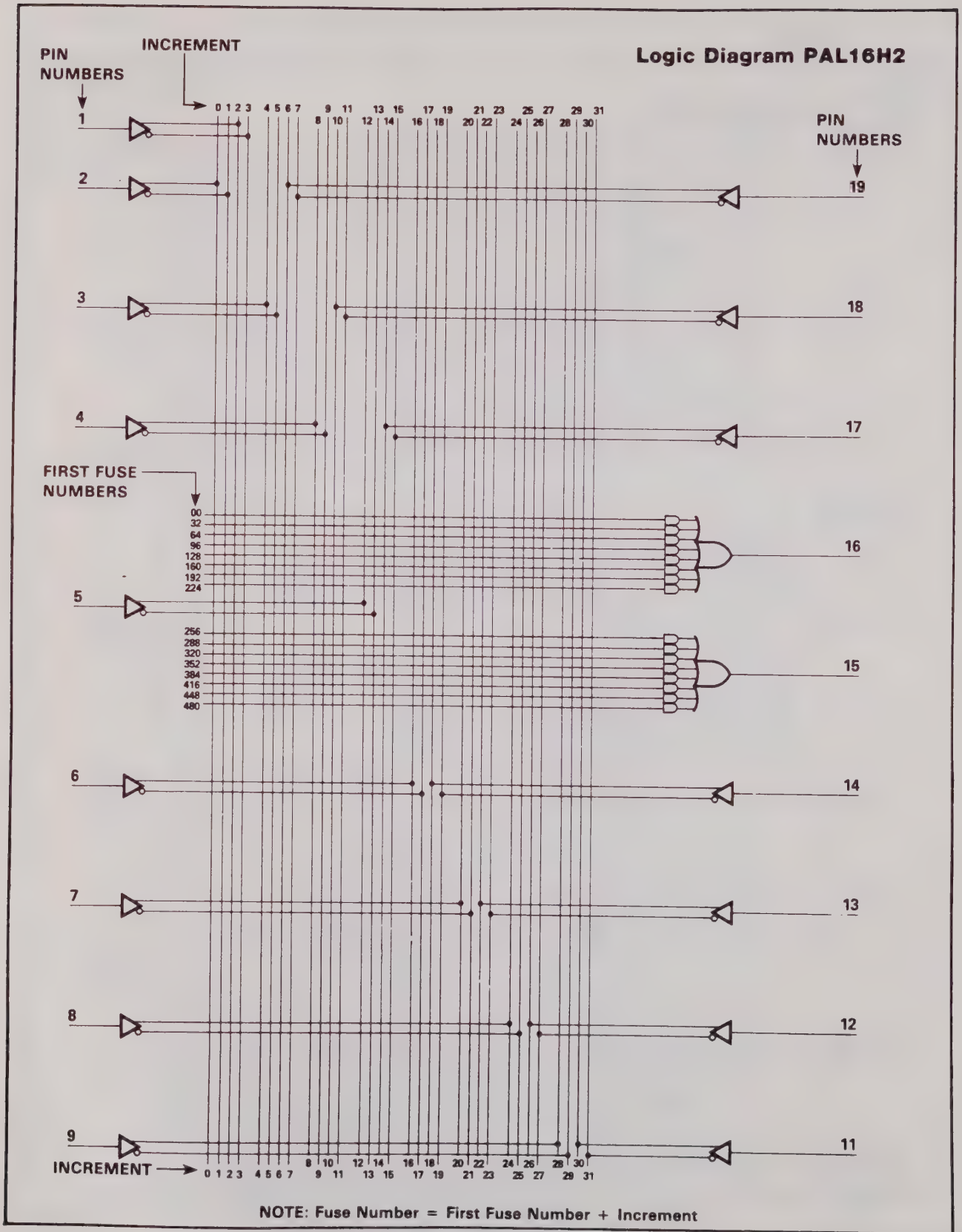
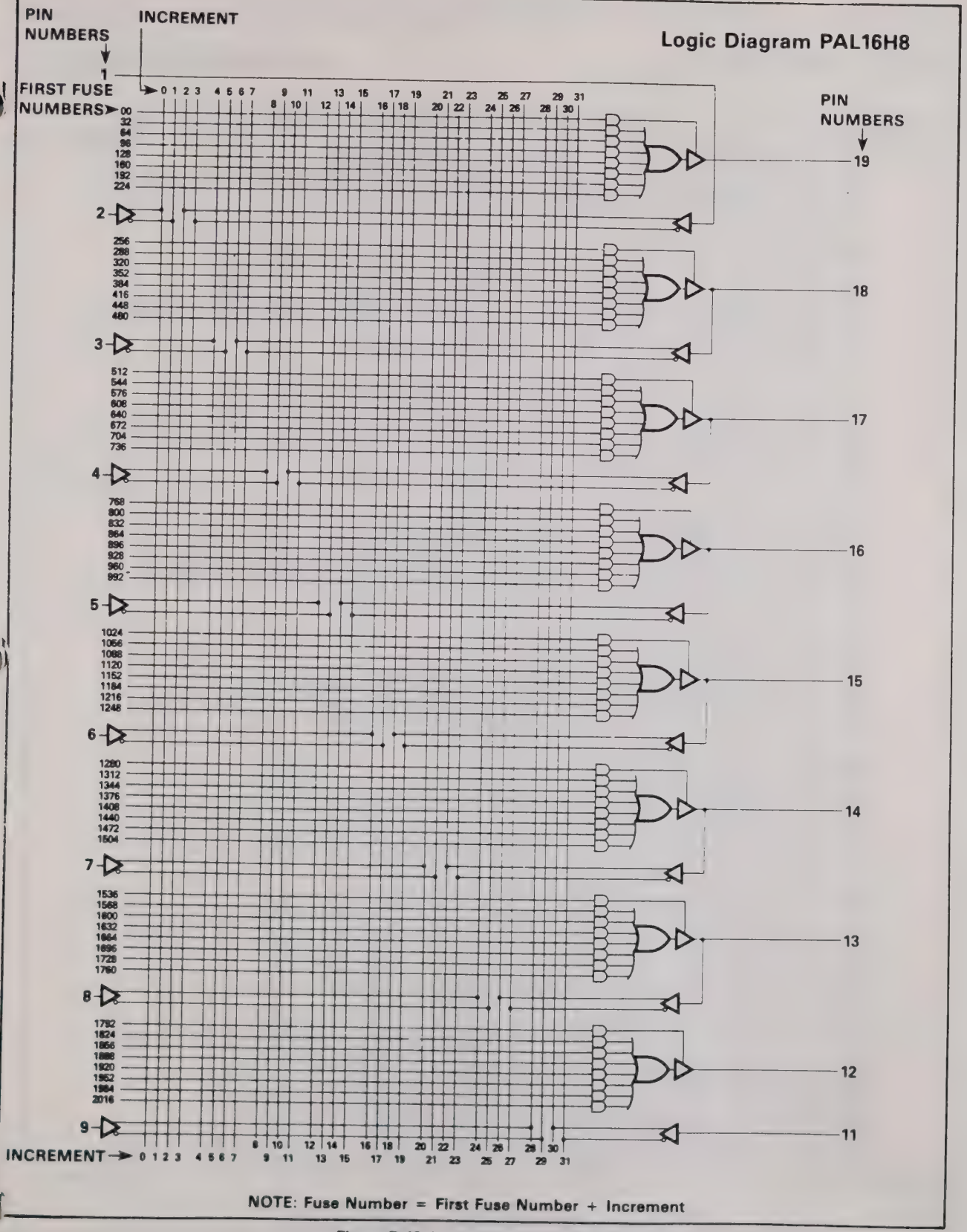


Figure B-9. Logic Diagram PAL16H2



NOTE: Fuse Number = First Fuse Number + Increment

Figure B-10. Logic Diagram PAL16H8

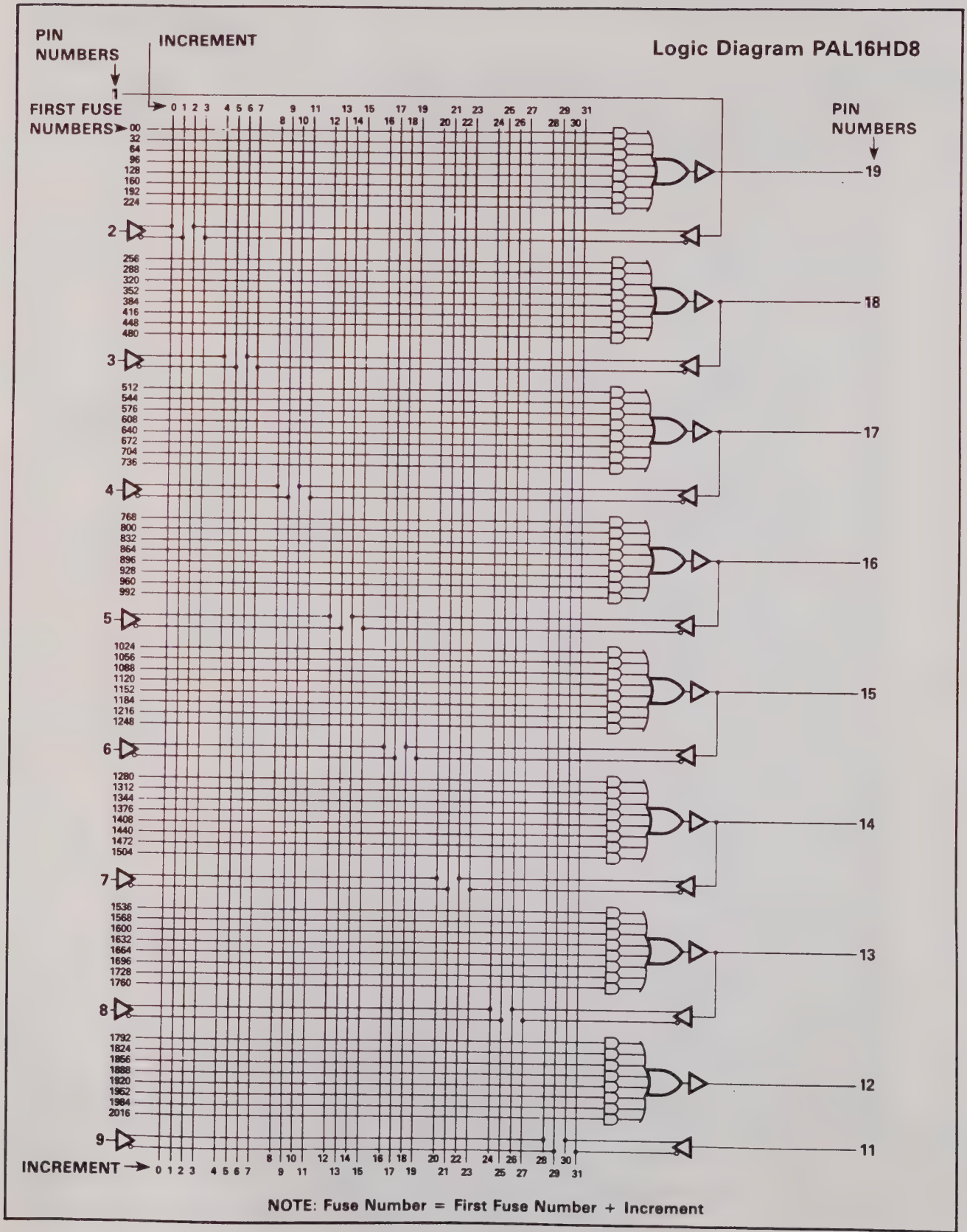


Figure B-11. Logic Diagram PAL16HD8

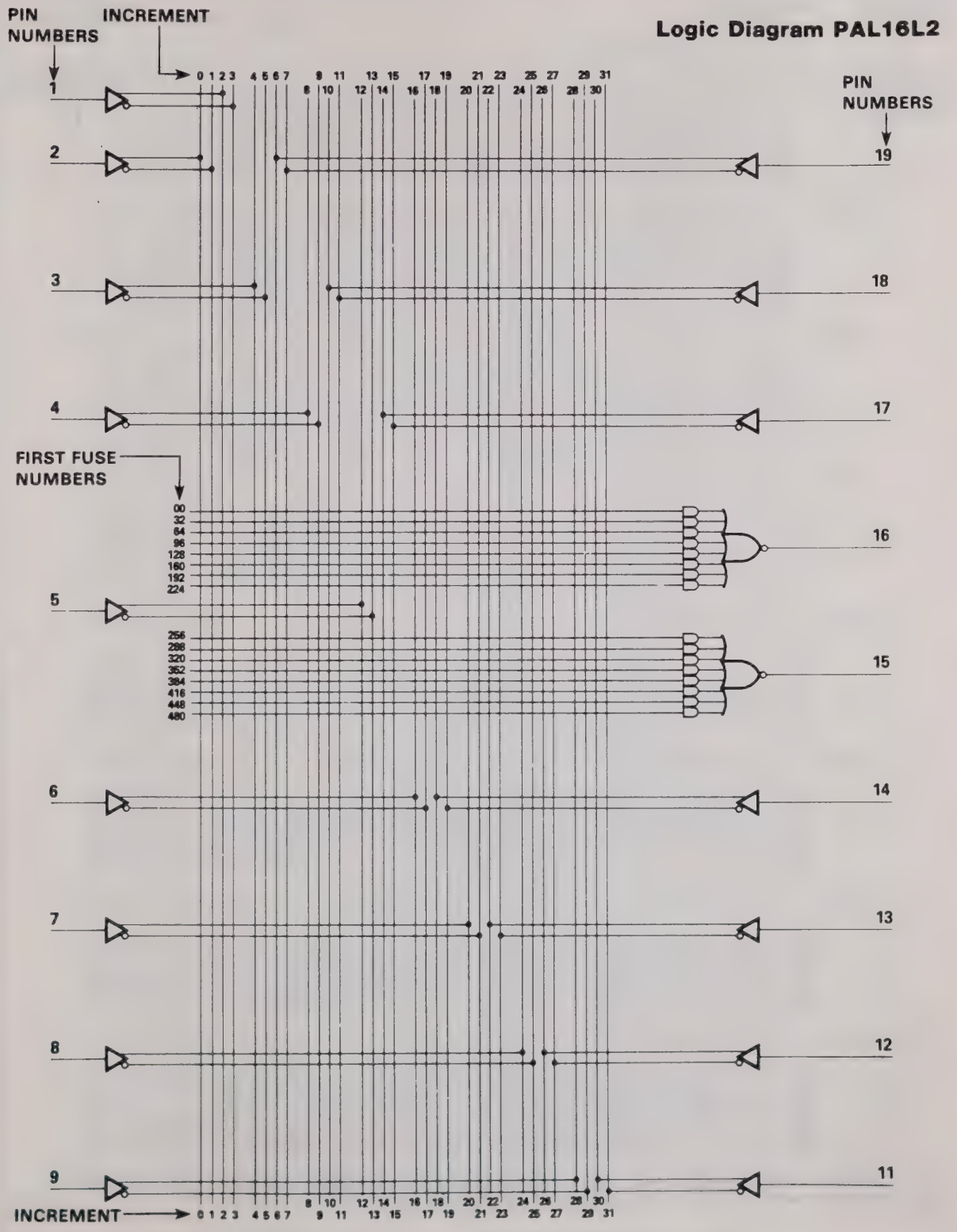


Figure B-12. Logic Diagram PAL16L2

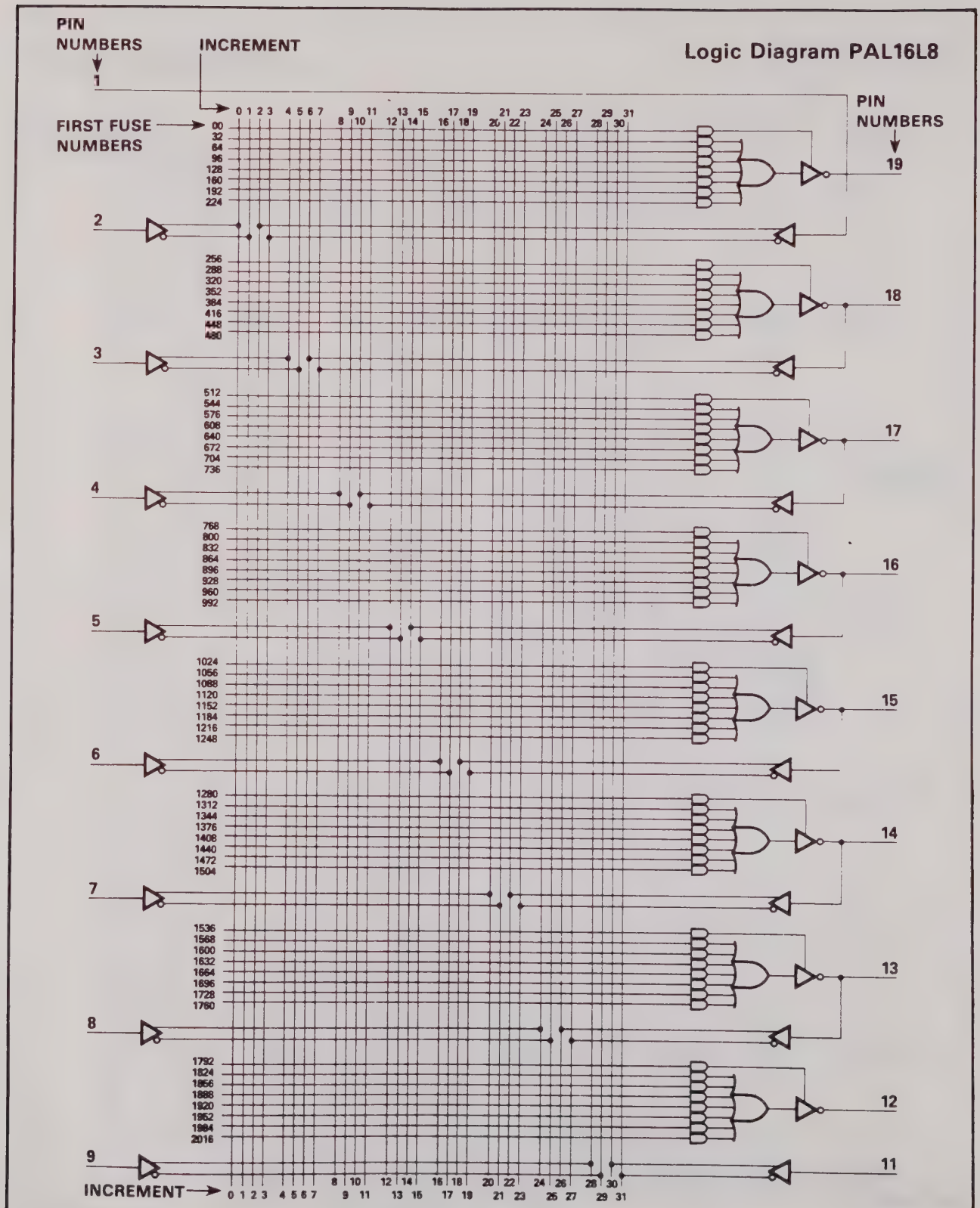
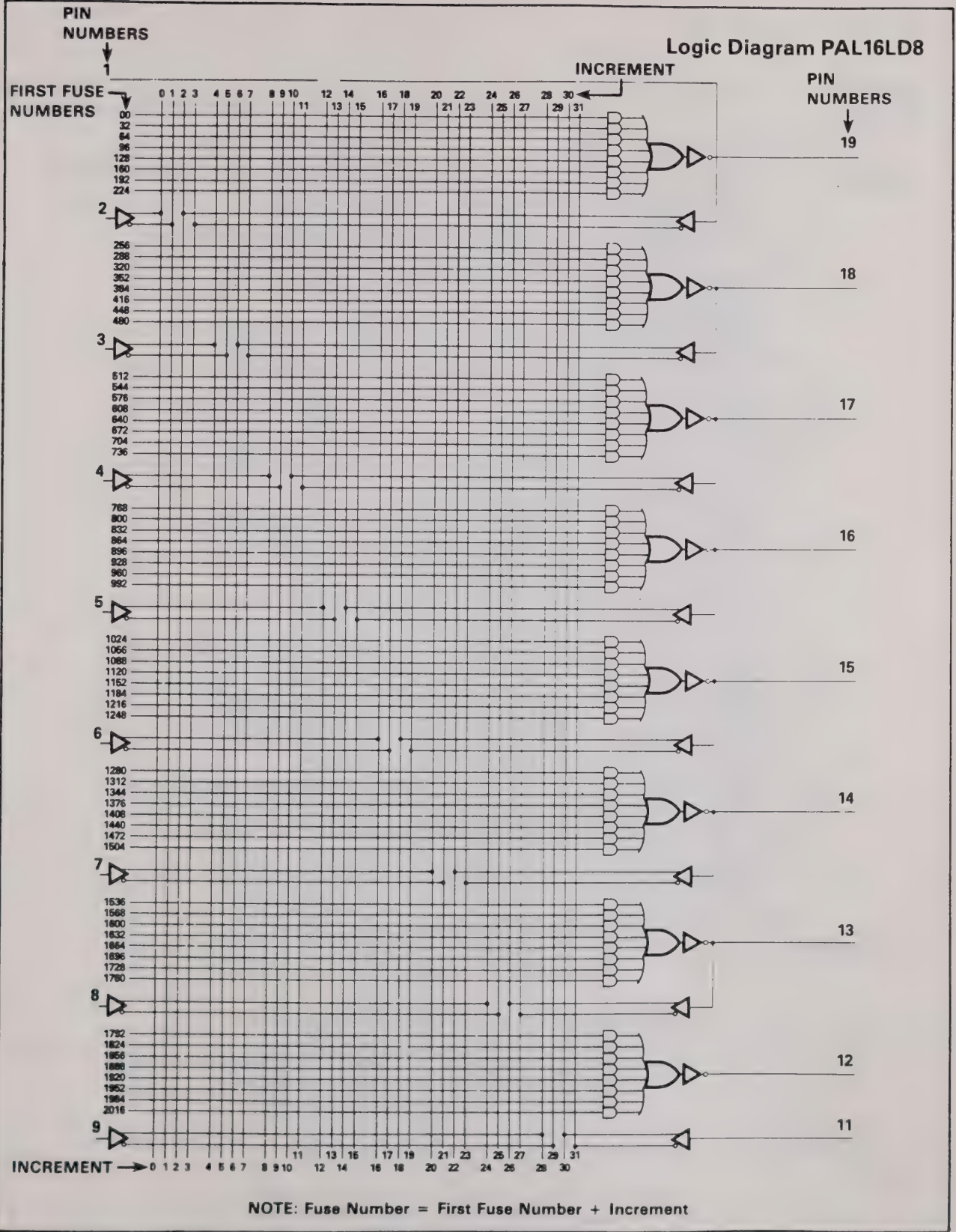


Figure B-13. Logic Diagram PAL16L8



NOTE: Fuse Number = First Fuse Number + Increment

Figure B-14. Logic Diagram PAL16LD8

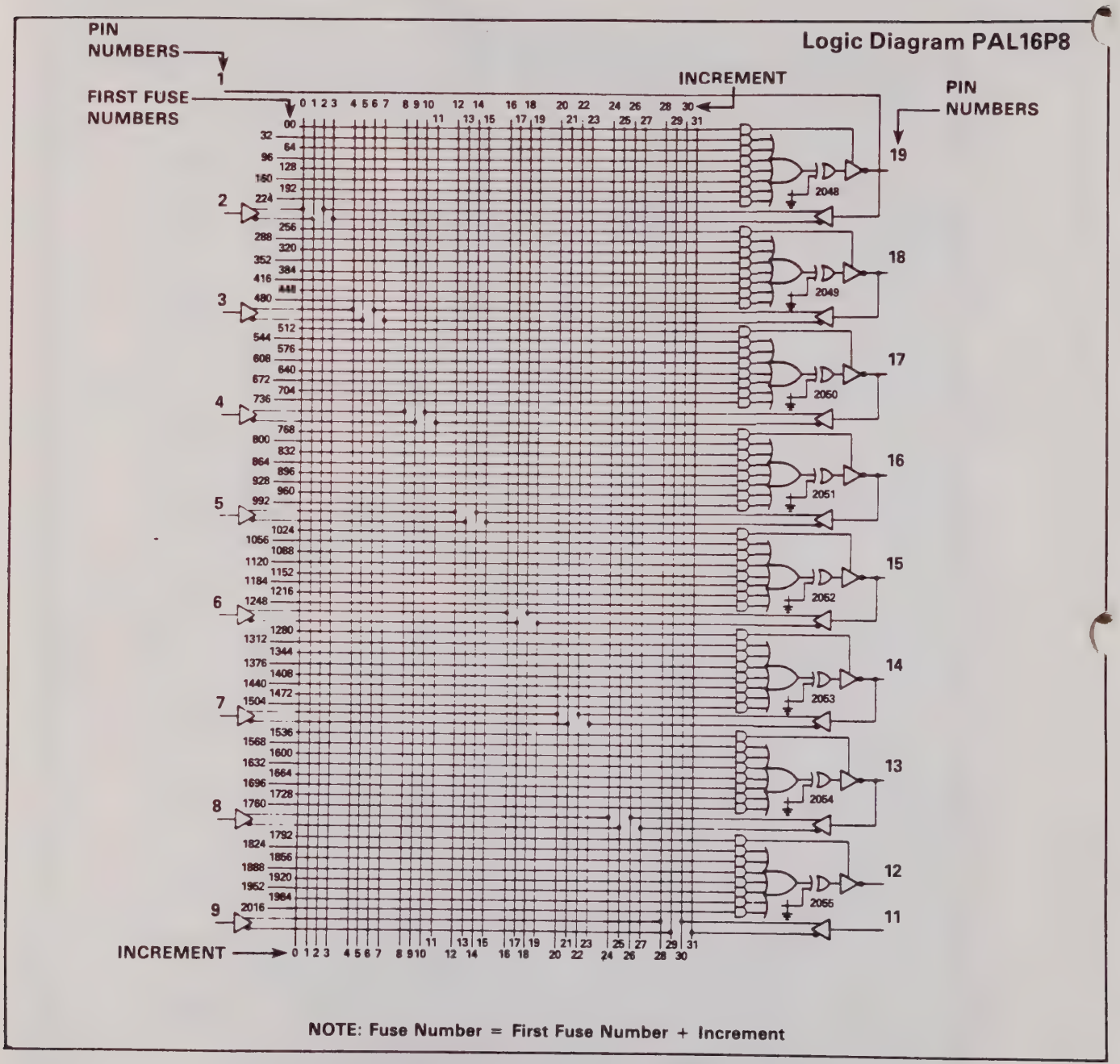


Figure B-15. Logic Diagram PAL16P8

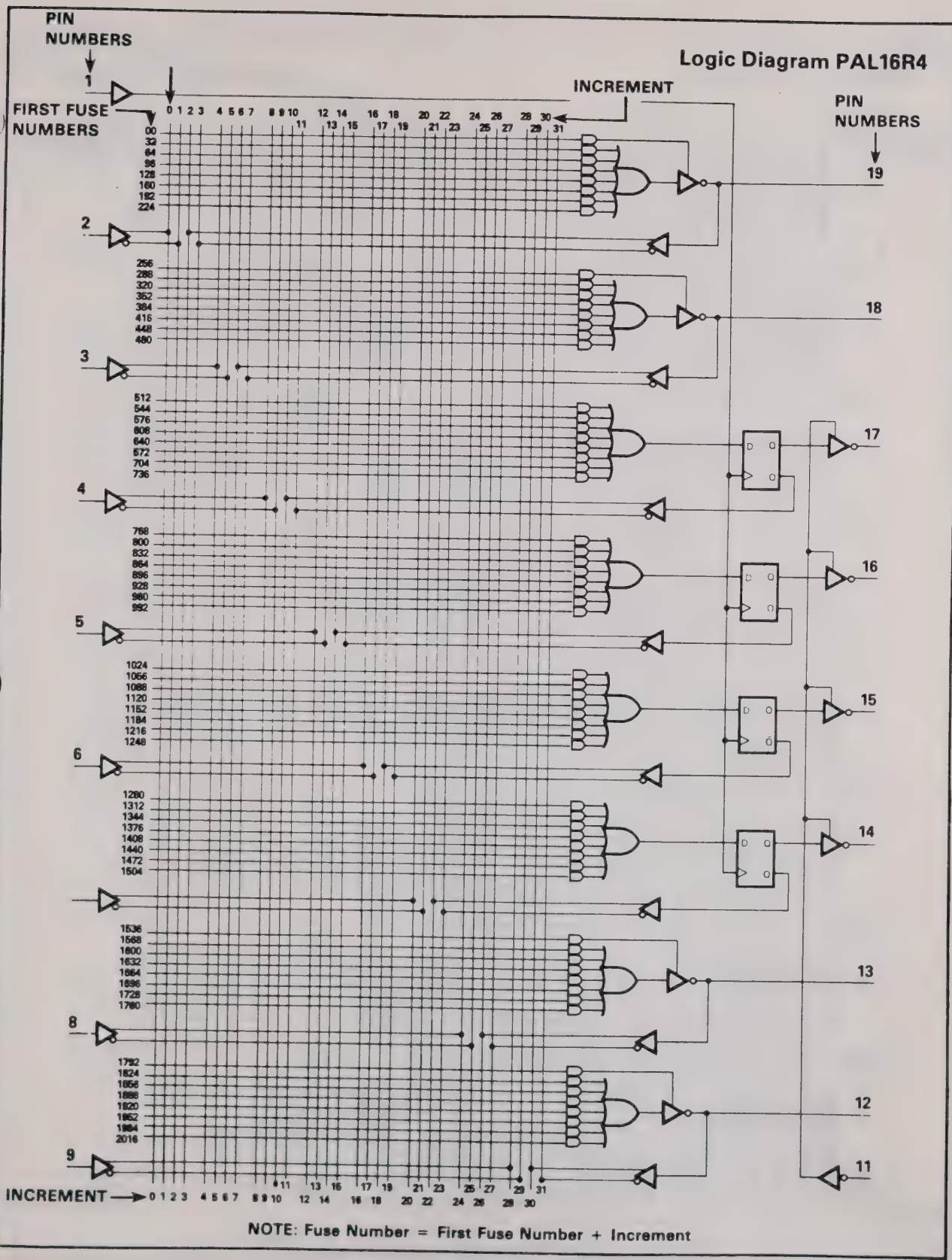


Figure B-16. Logic Diagram PAL16R4

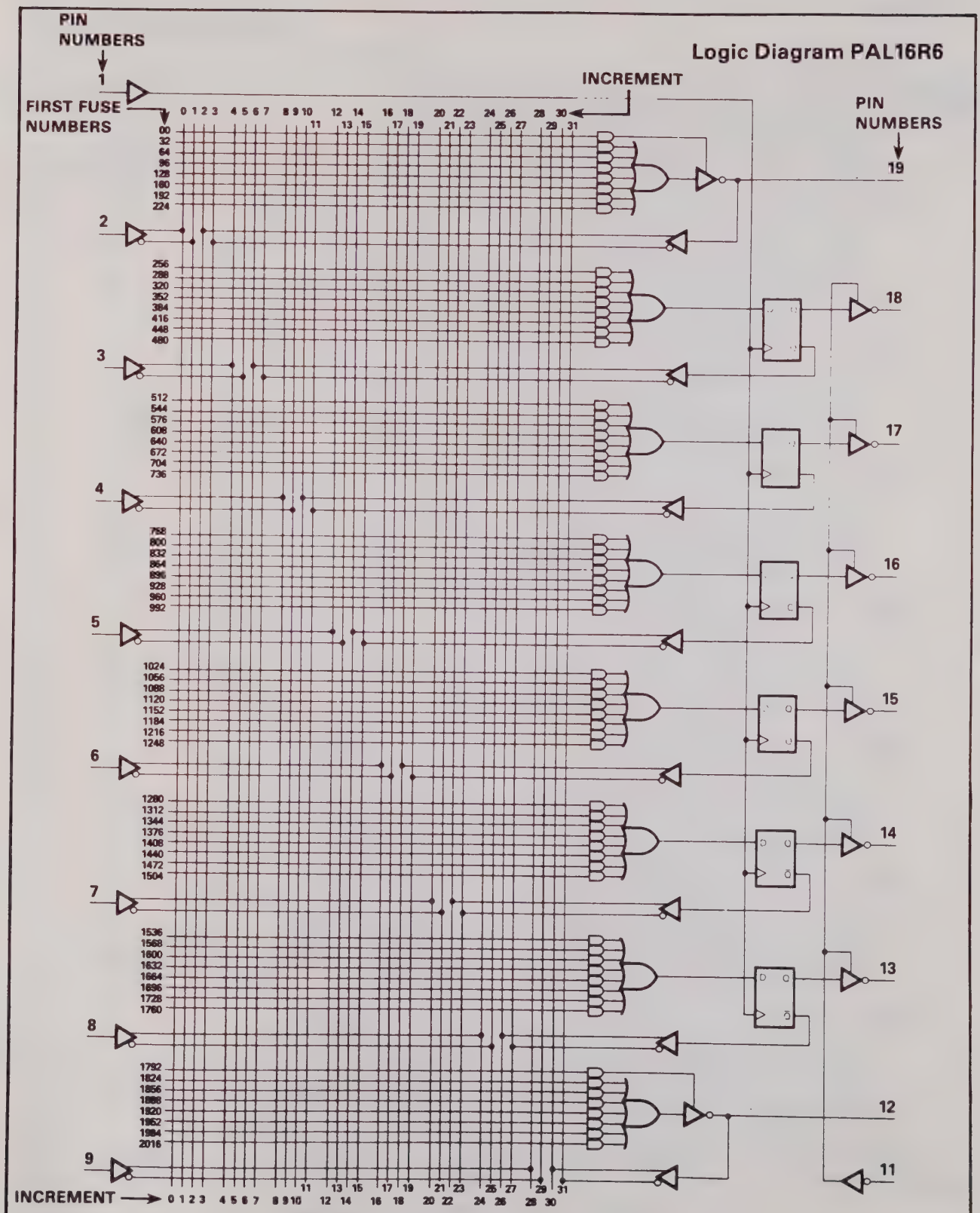


Figure B-17. Logic Diagram PAL16R6

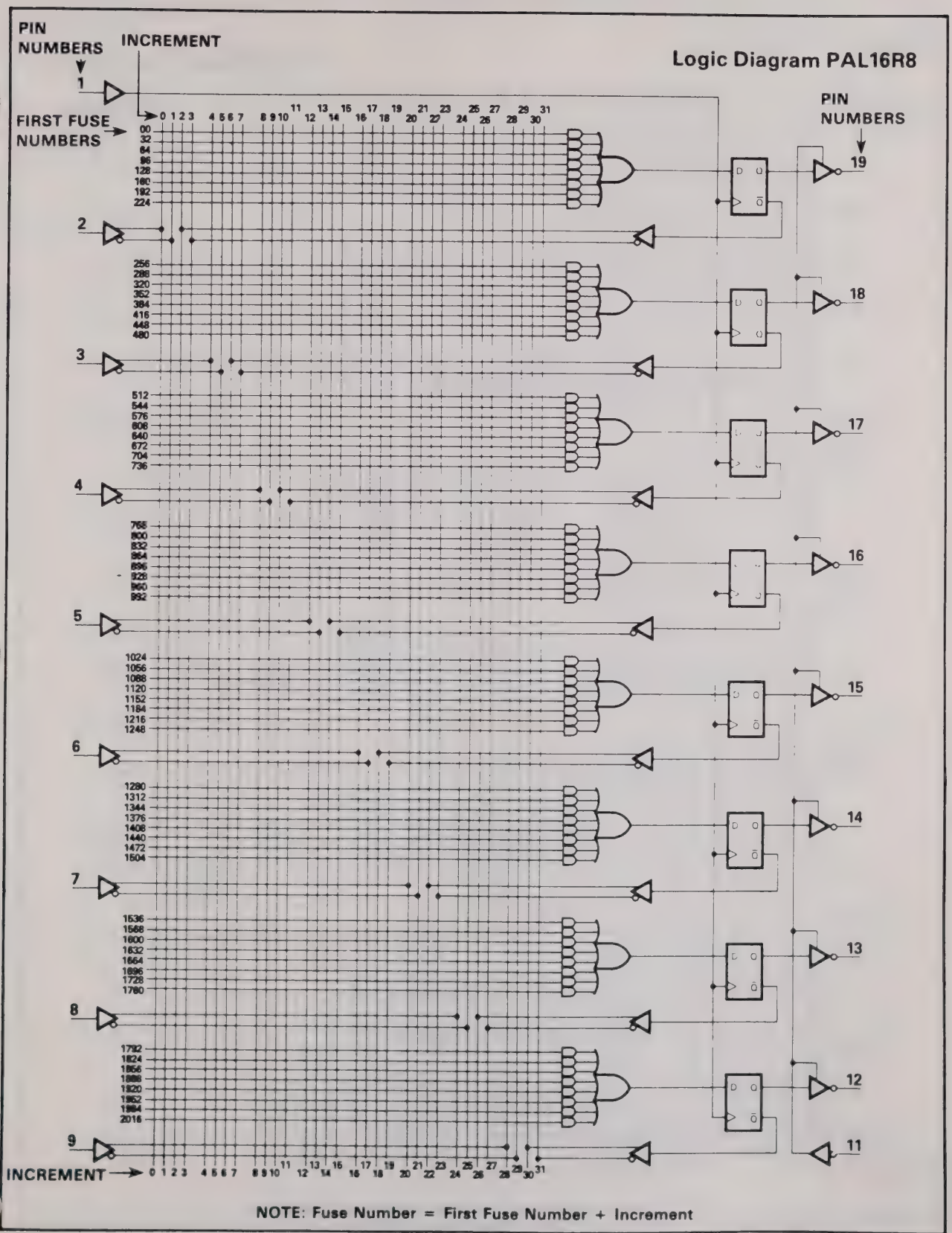


Figure B-18. Logic Diagram PAL16R8

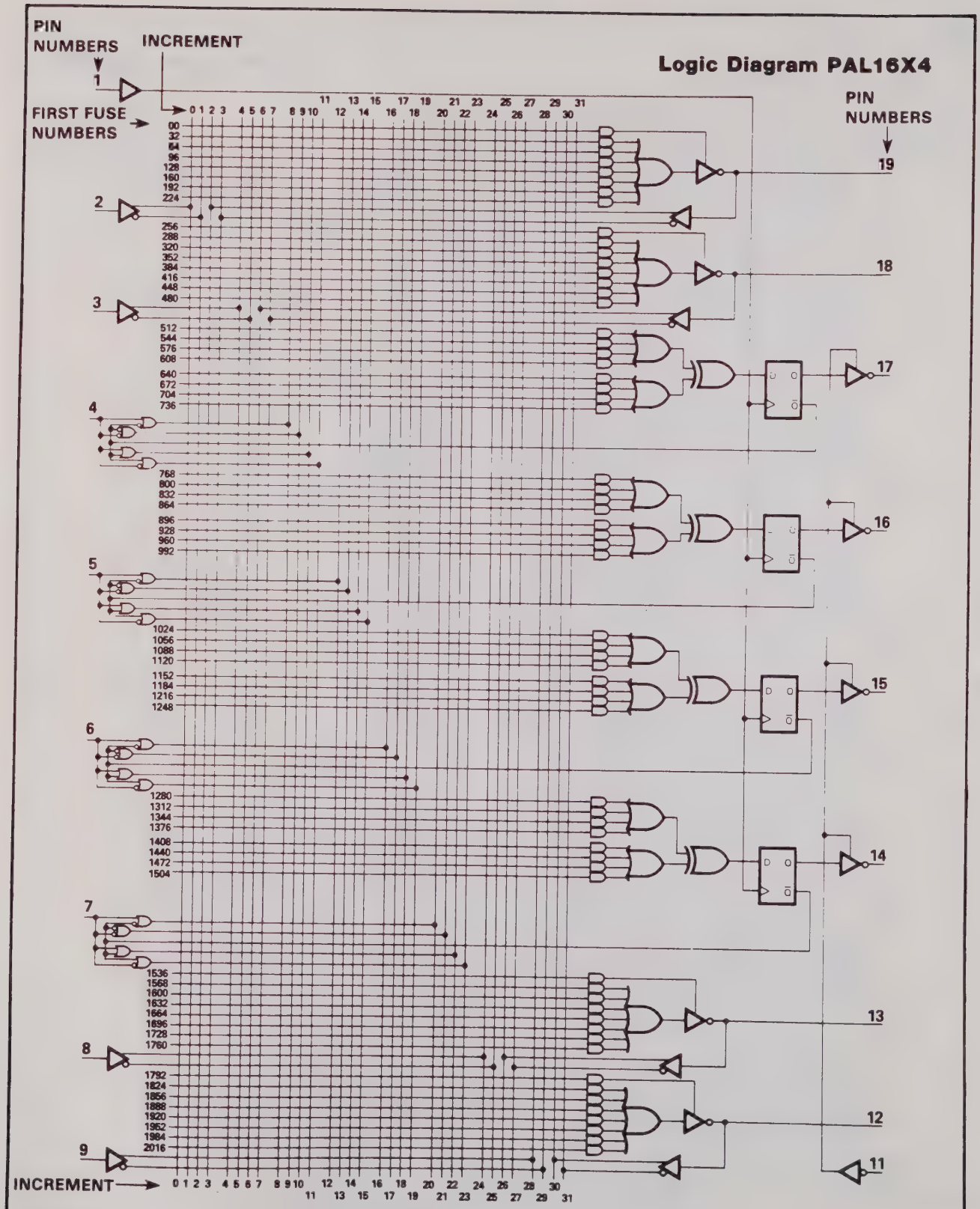


Figure B-19. Logic Diagram PAL16X4

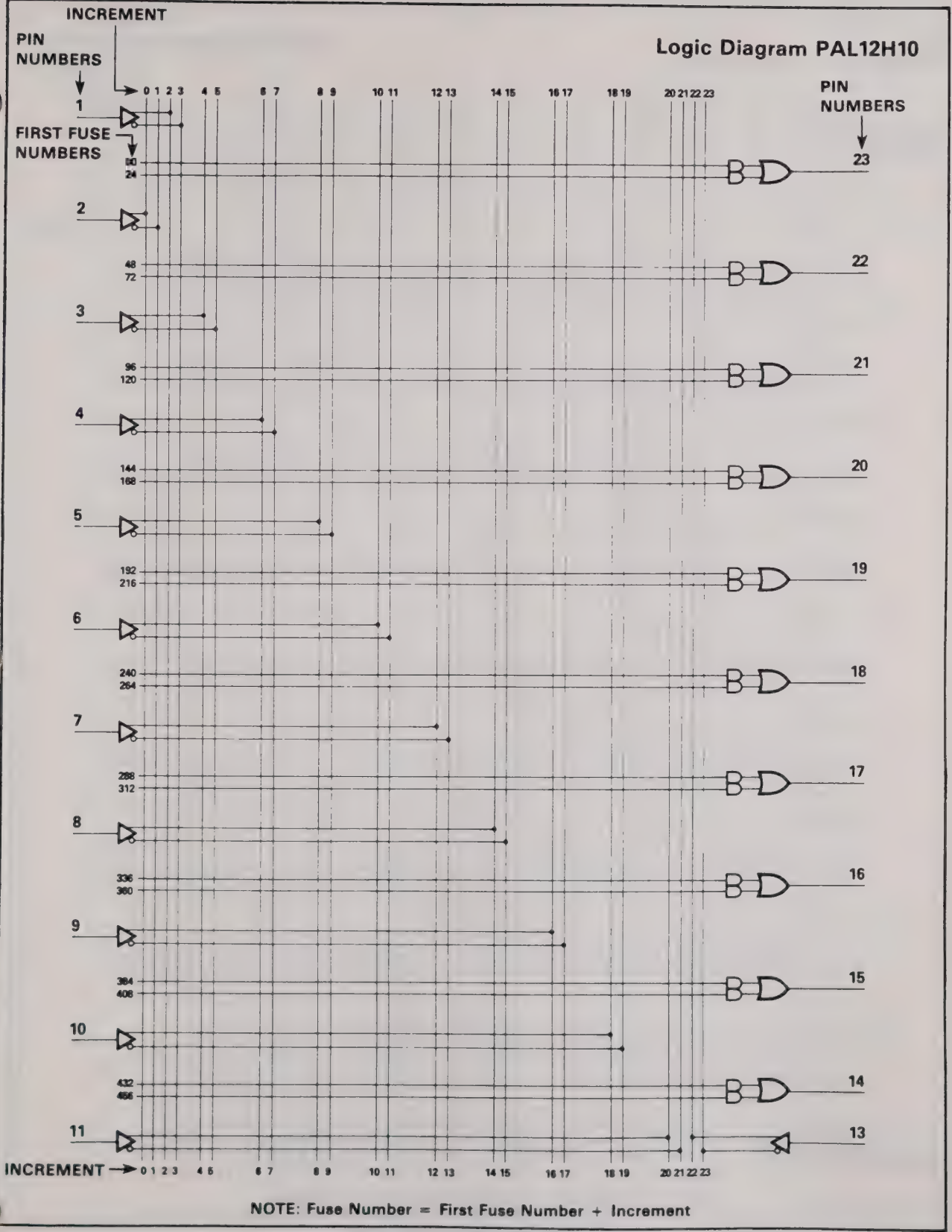


Figure B-20. Logic Diagram PAL12H10

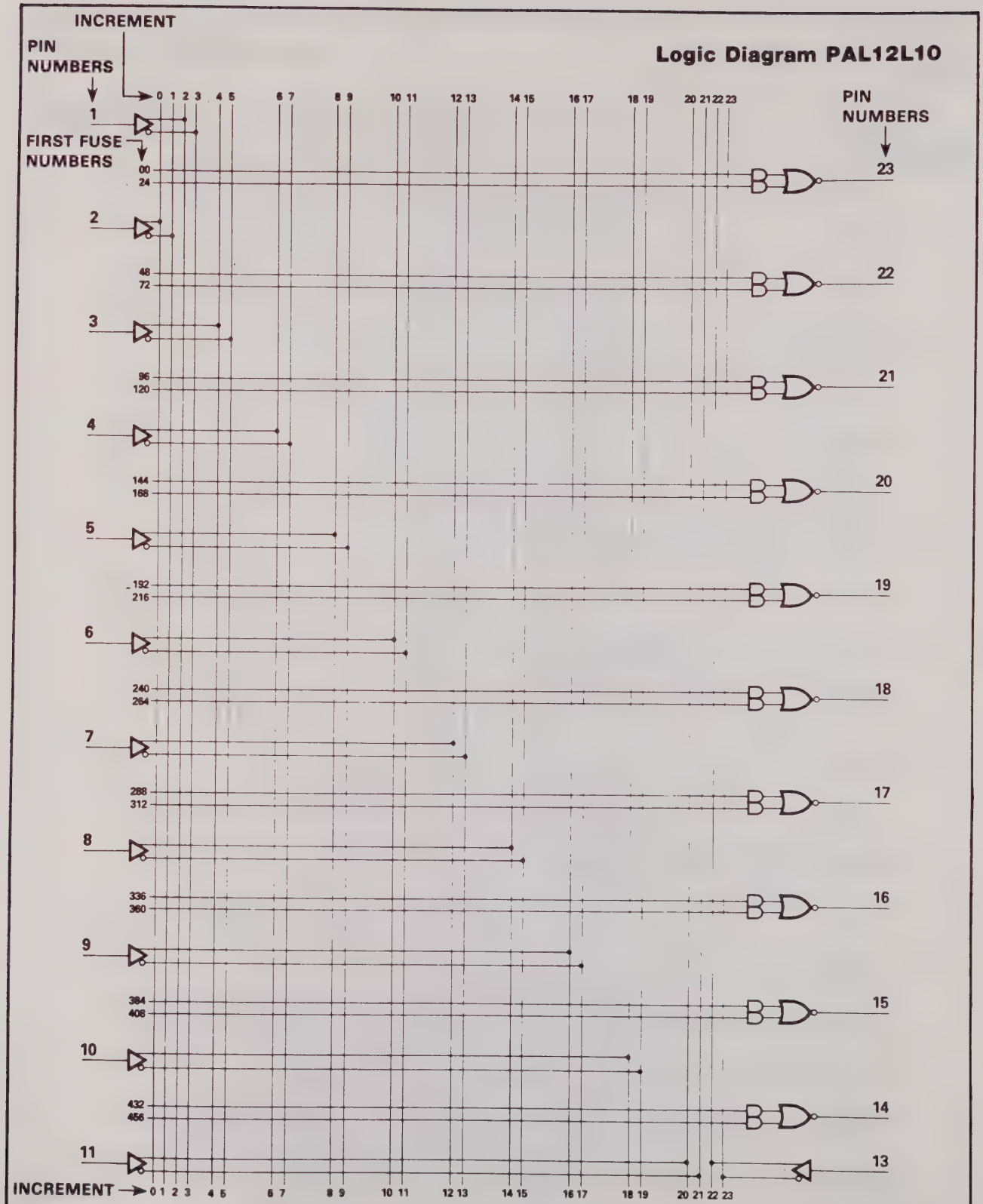


Figure B-21. Logic Diagram PAL12L10

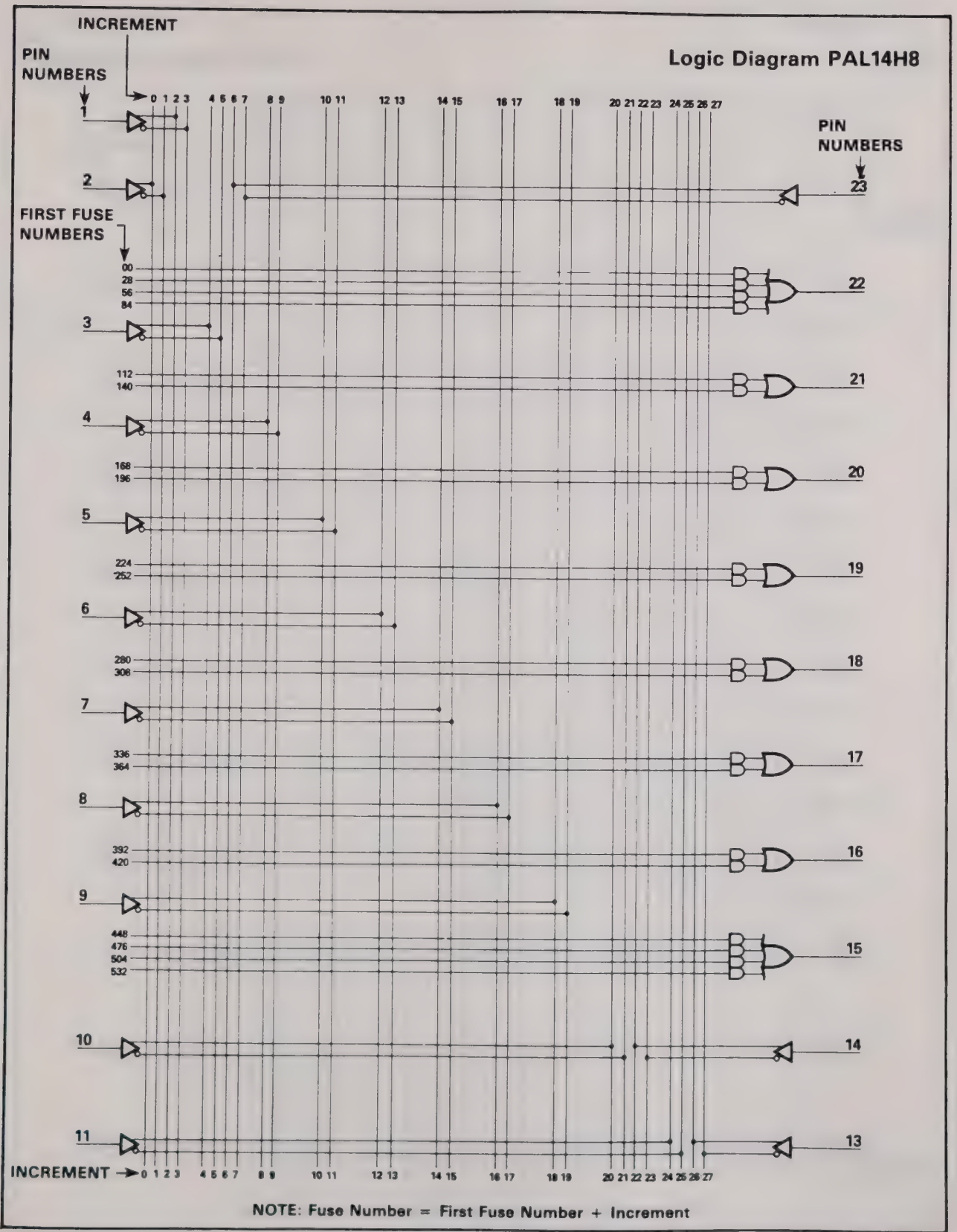


Figure B-22. Logic Diagram PAL14H8

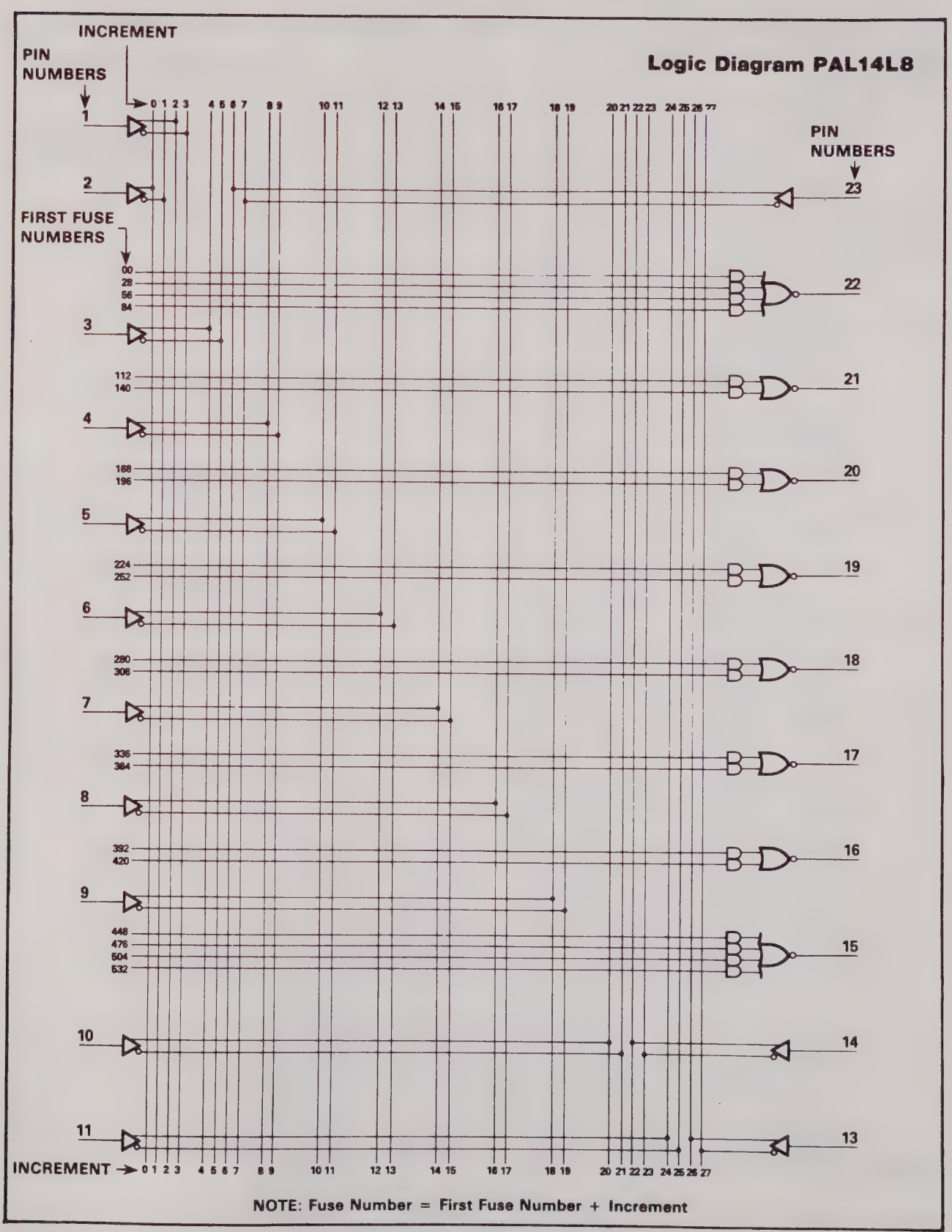


Figure B-23. Logic Diagram PAL14L8

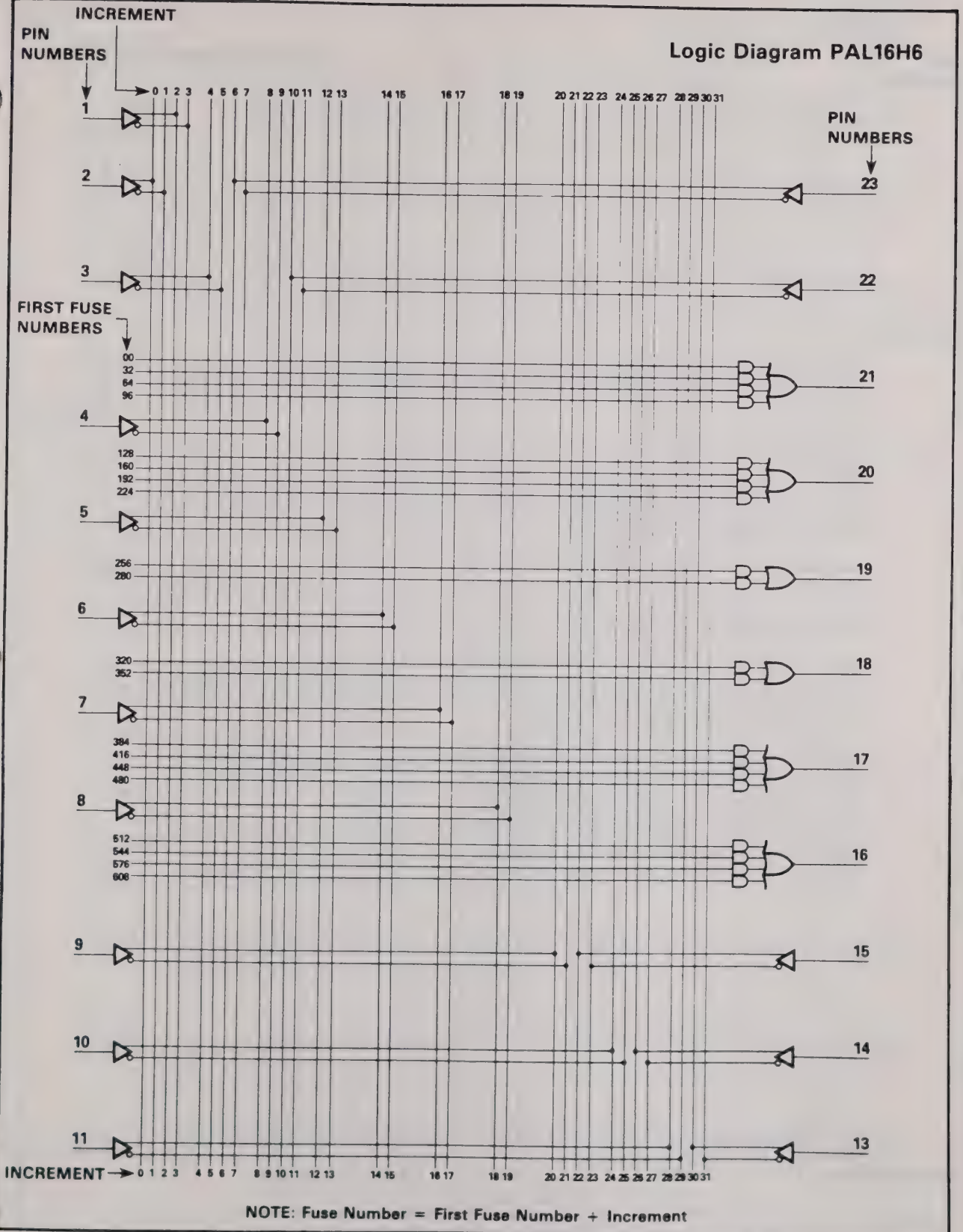
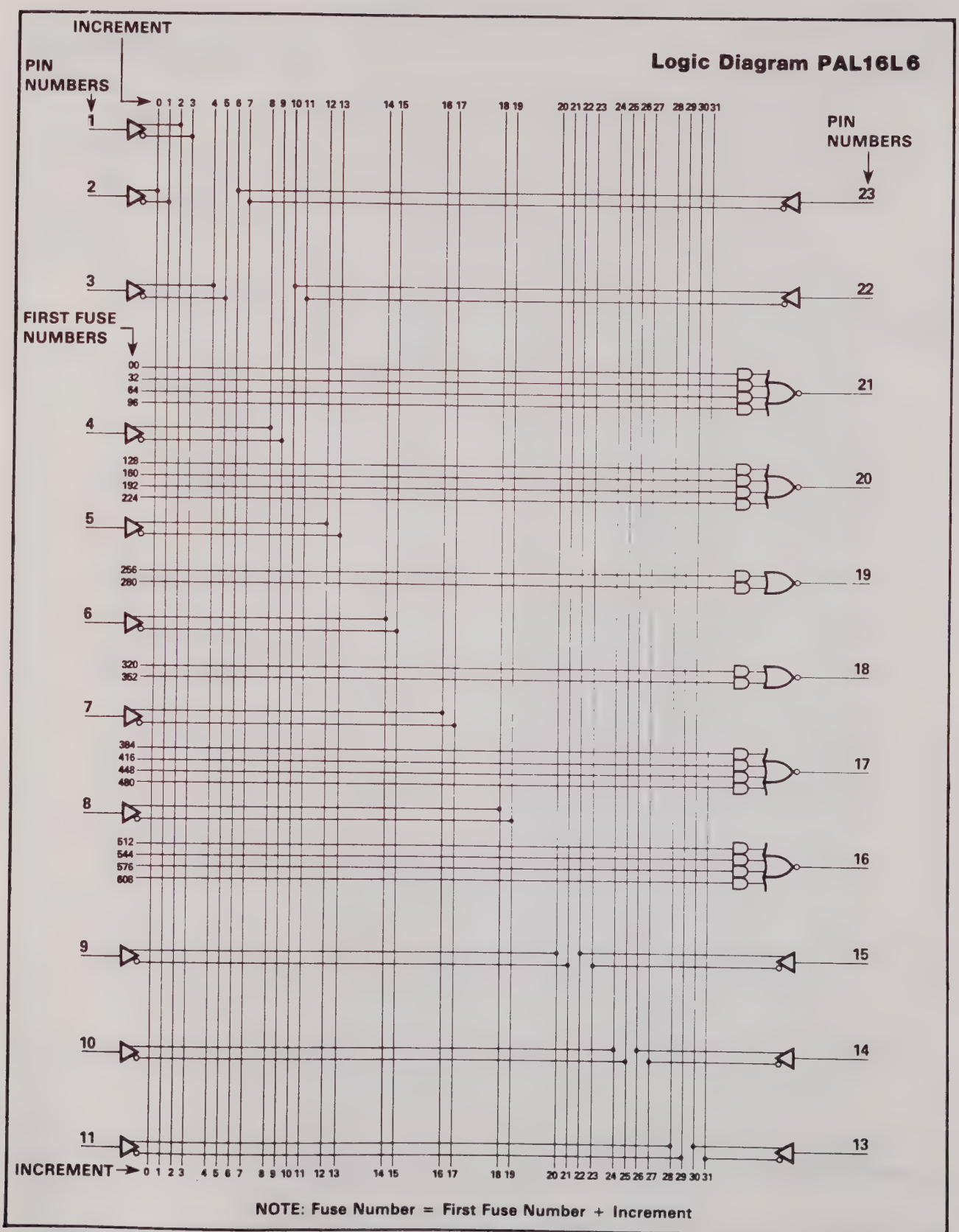


Figure B-24. Logic Diagram PAL16H6



NOTE: Fuse Number = First Fuse Number + Increment

Figure B-25. Logic Diagram PAL16L6

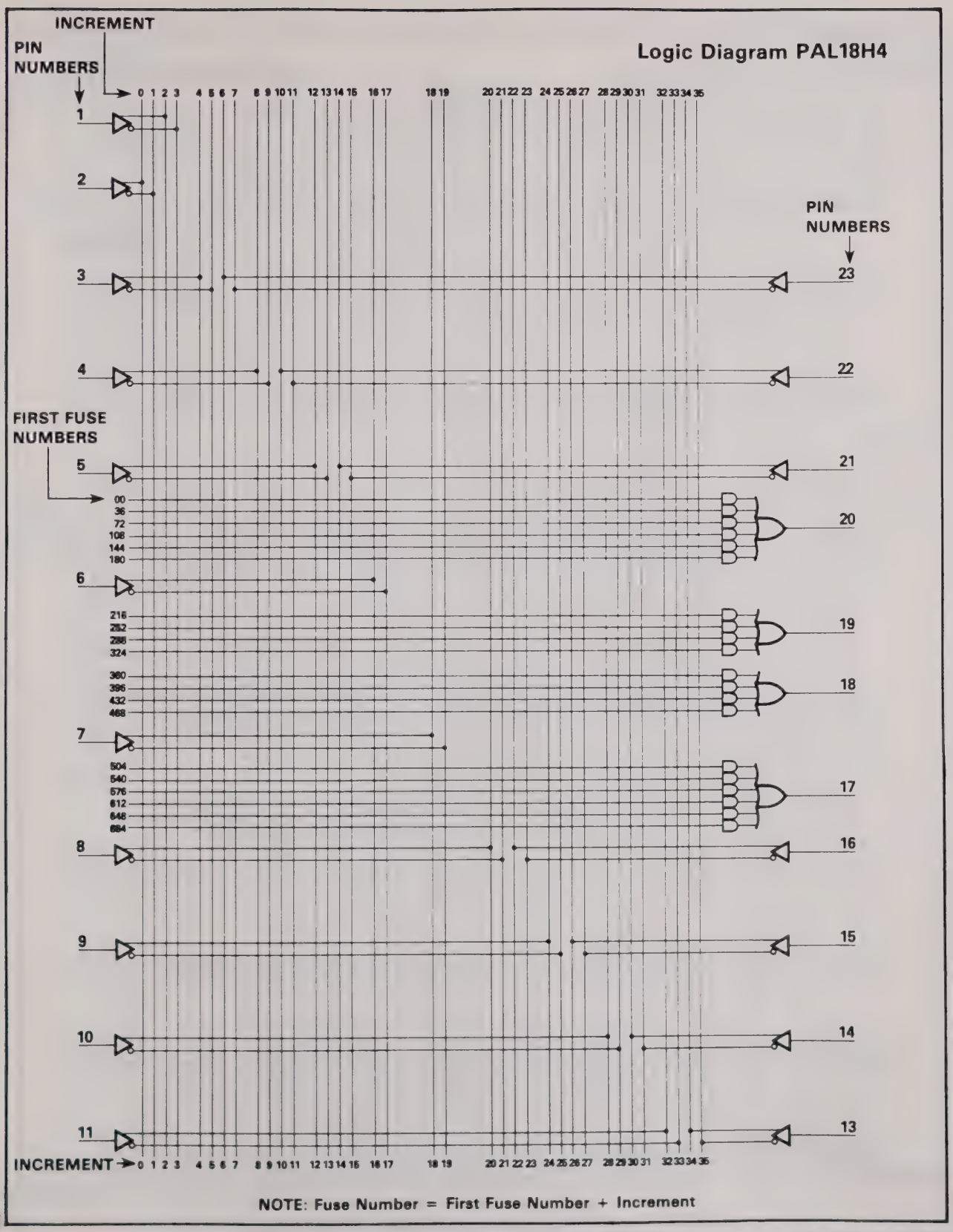


Figure B-26. Logic Diagram PAL18H4

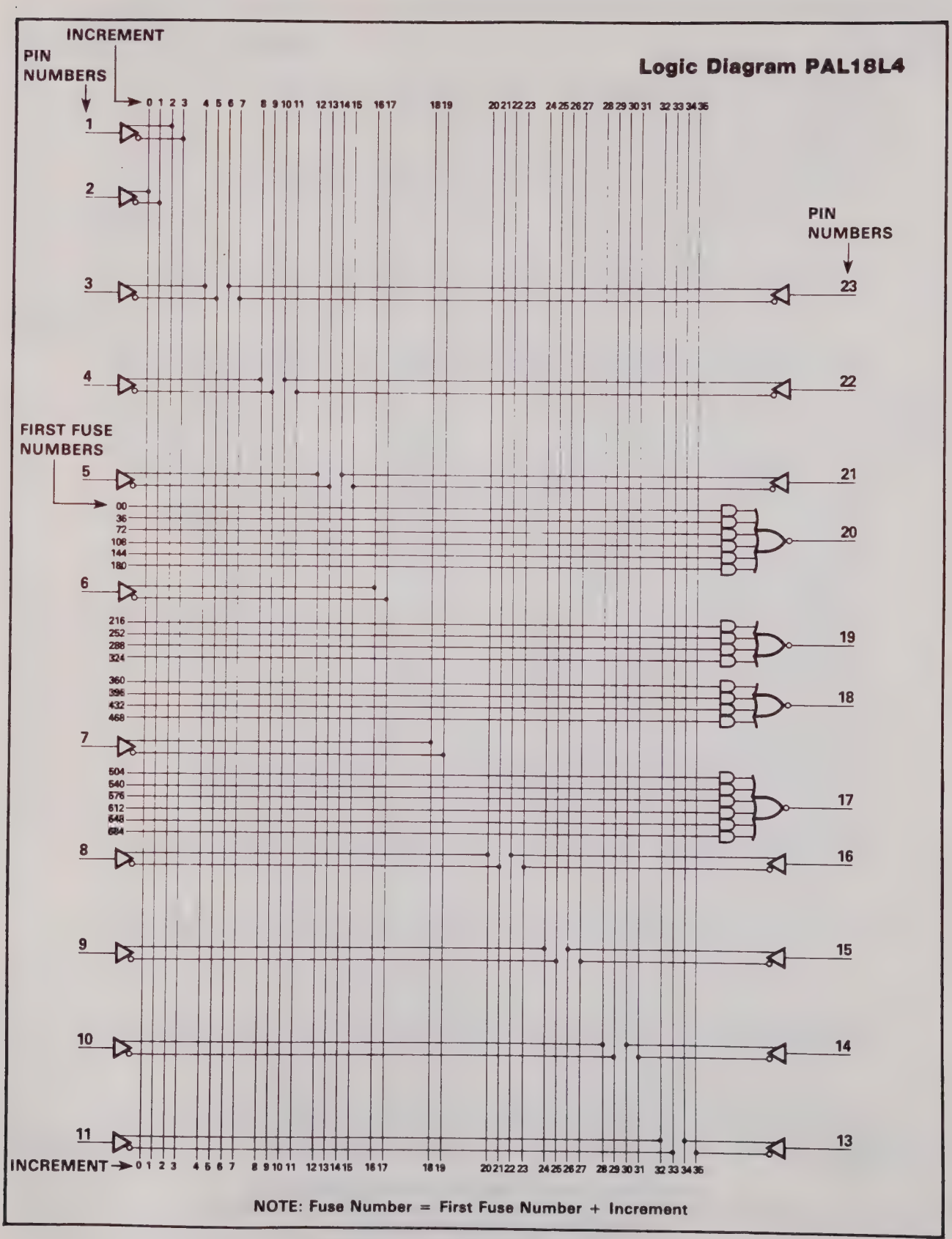


Figure B-27. Logic Diagram PAL18L4

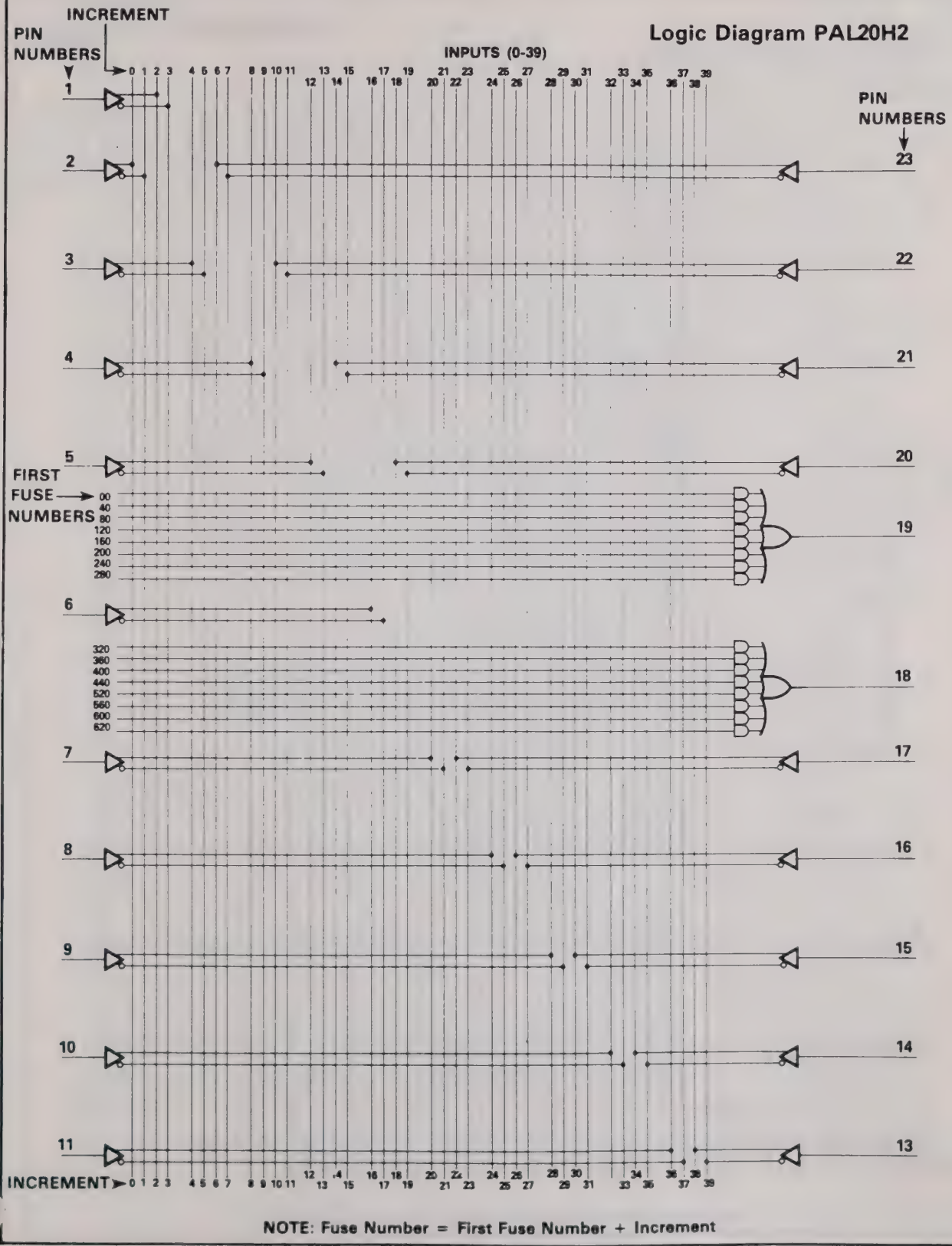


Figure B-28. Logic Diagram PAL20H2

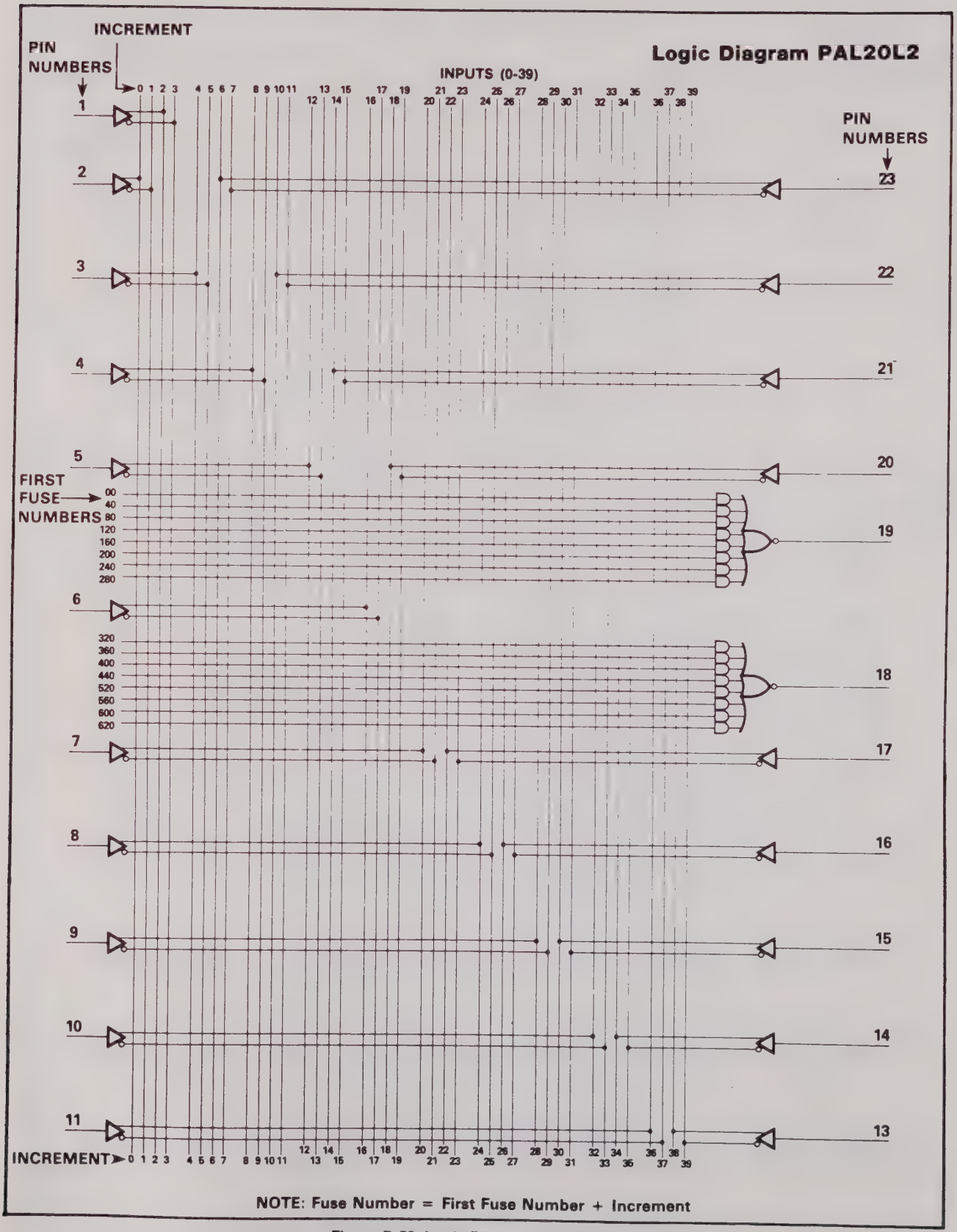


Figure B-29. Logic Diagram PAL20L2

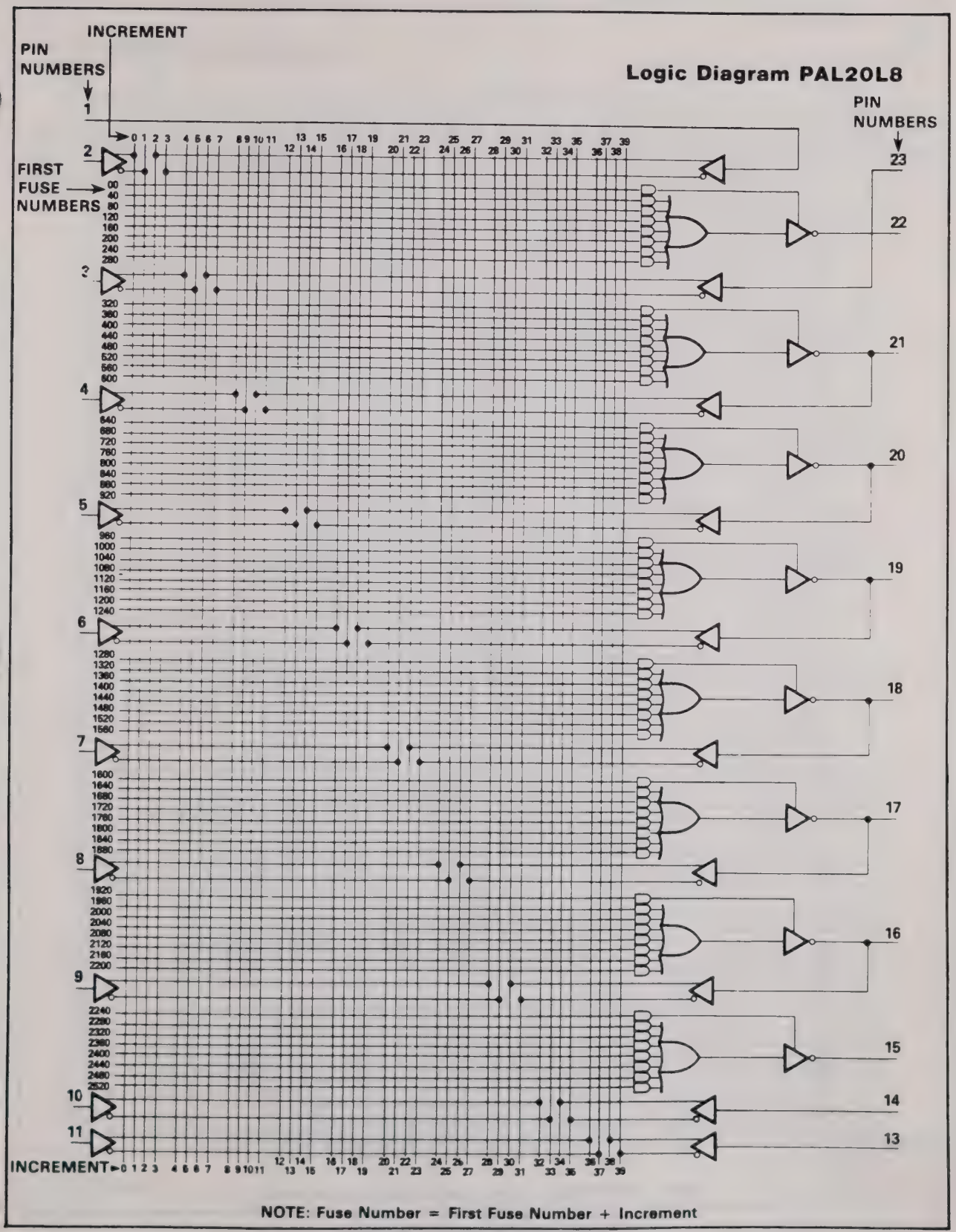


Figure B-30. Logic Diagram PAL20L8

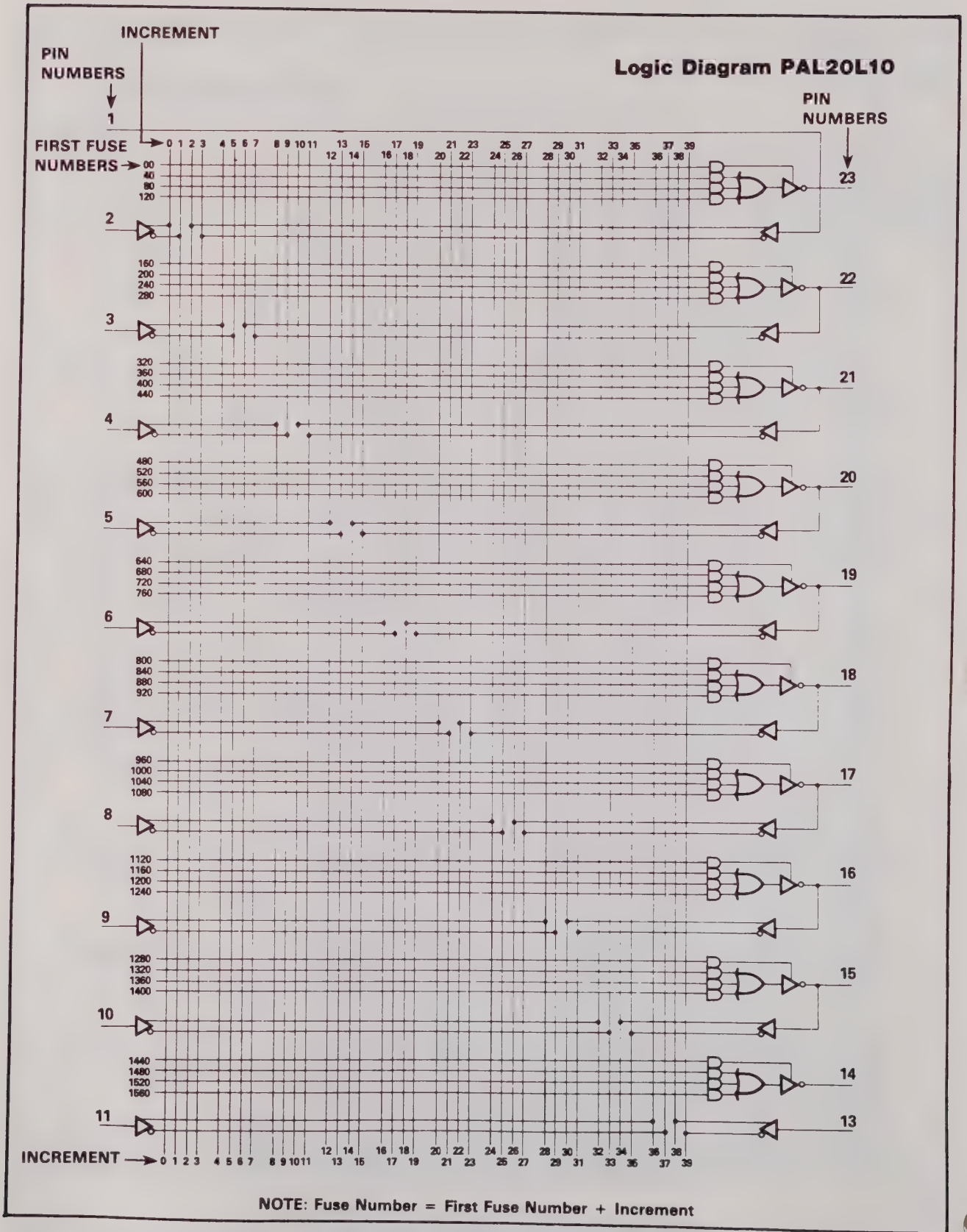


Figure B-31. Logic Diagram PAL20L10

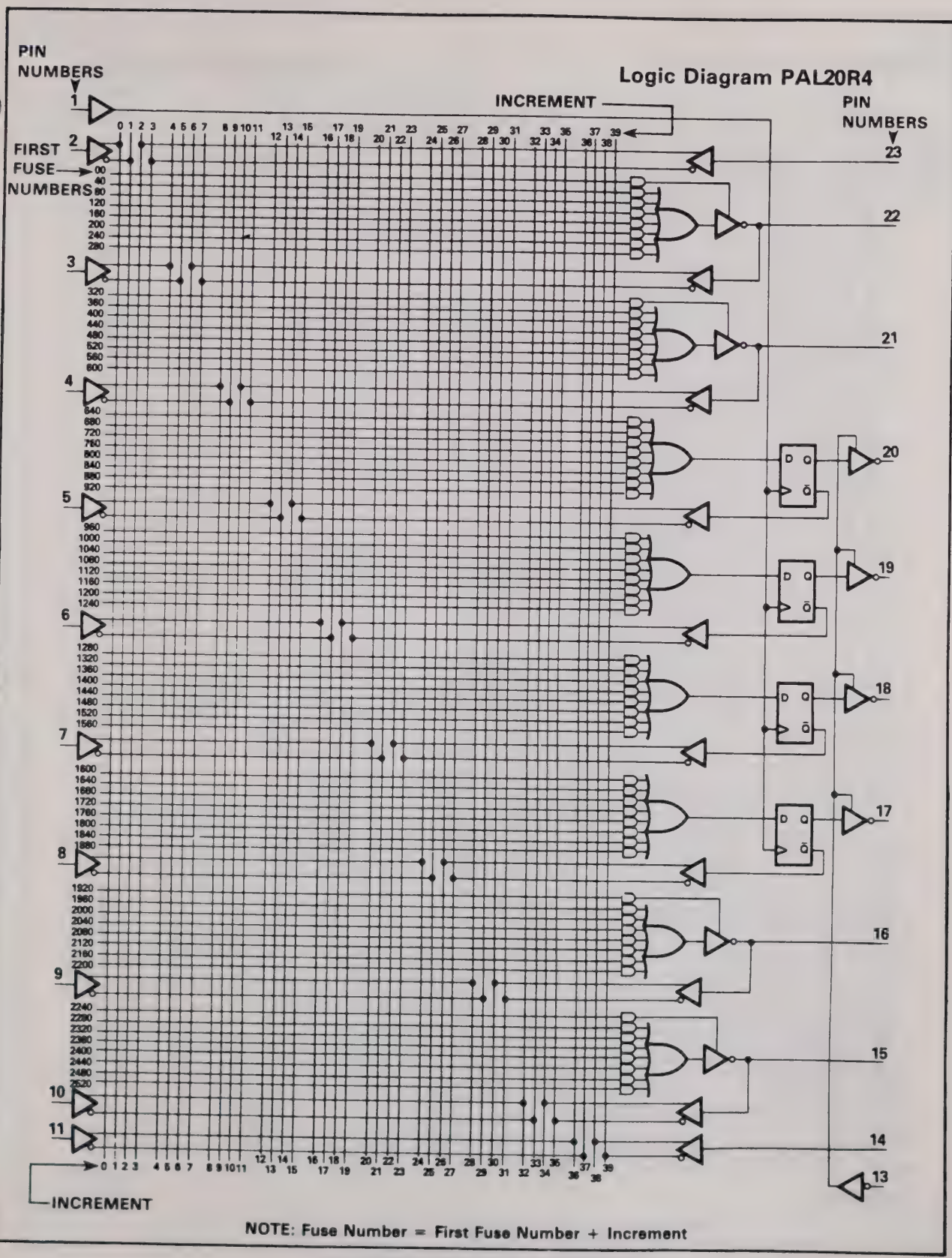


Figure B-32. Logic Diagram PAL20R4

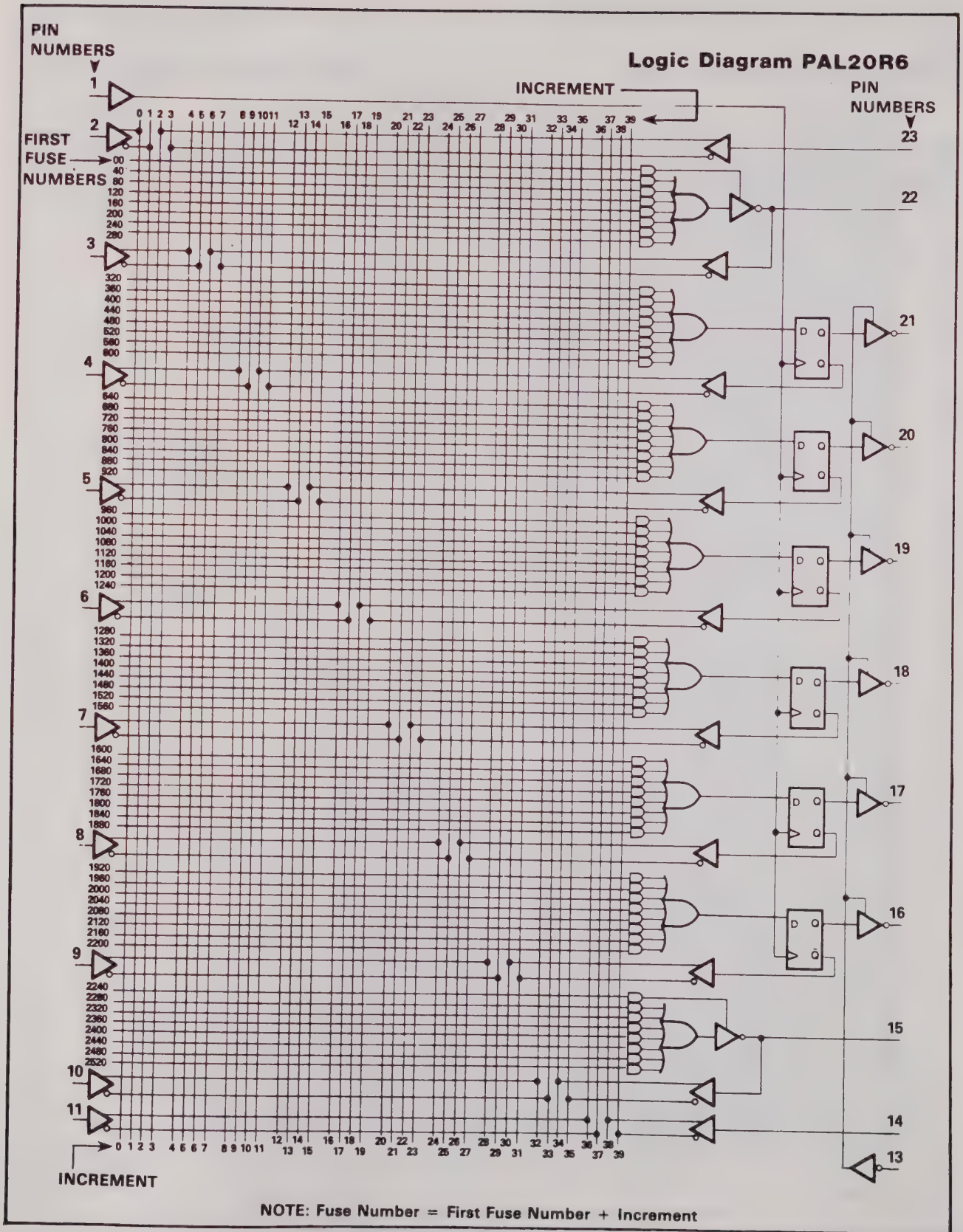


Figure B-33. Logic Diagram PAL20R6

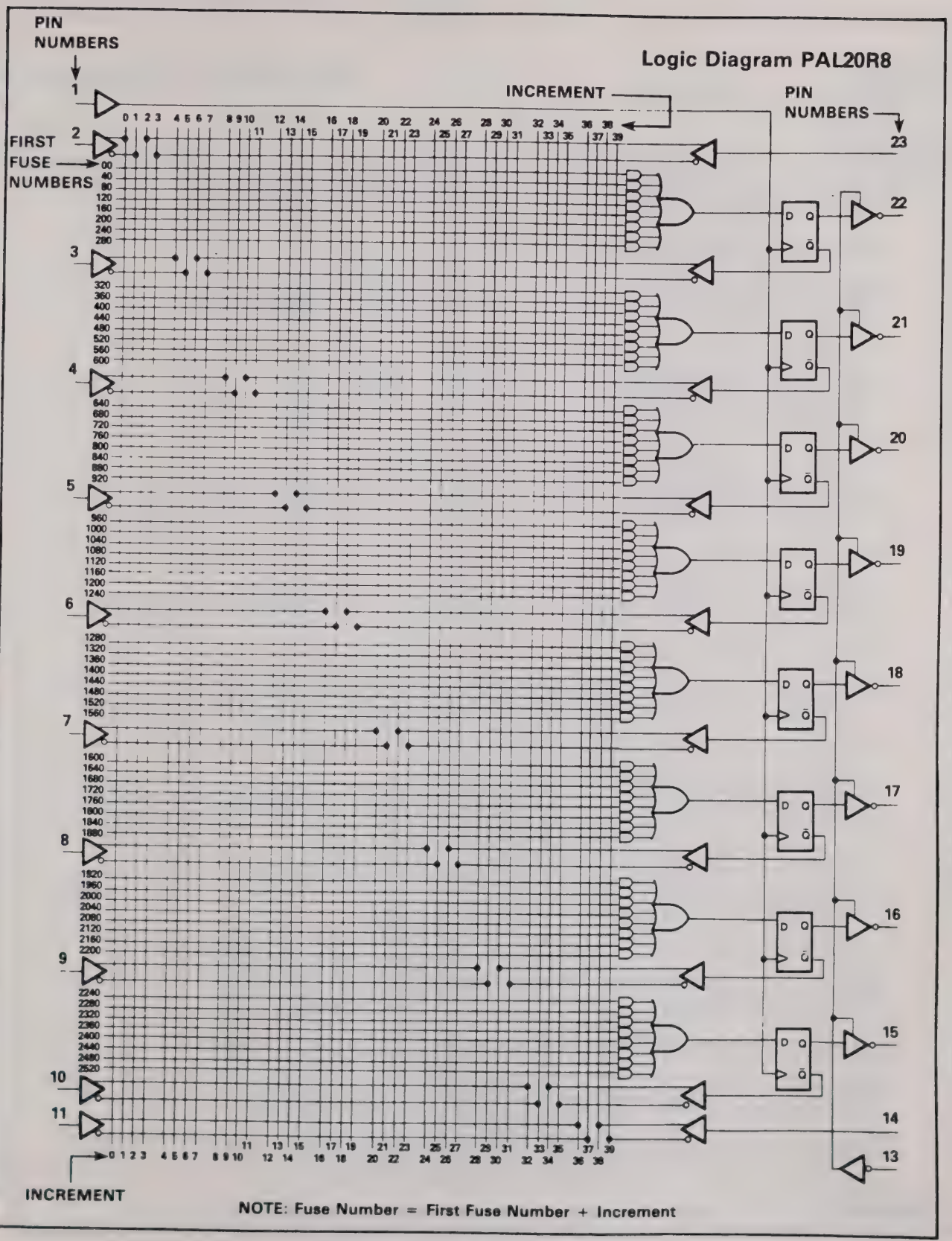


Figure B-34. Logic Diagram PAL20R8

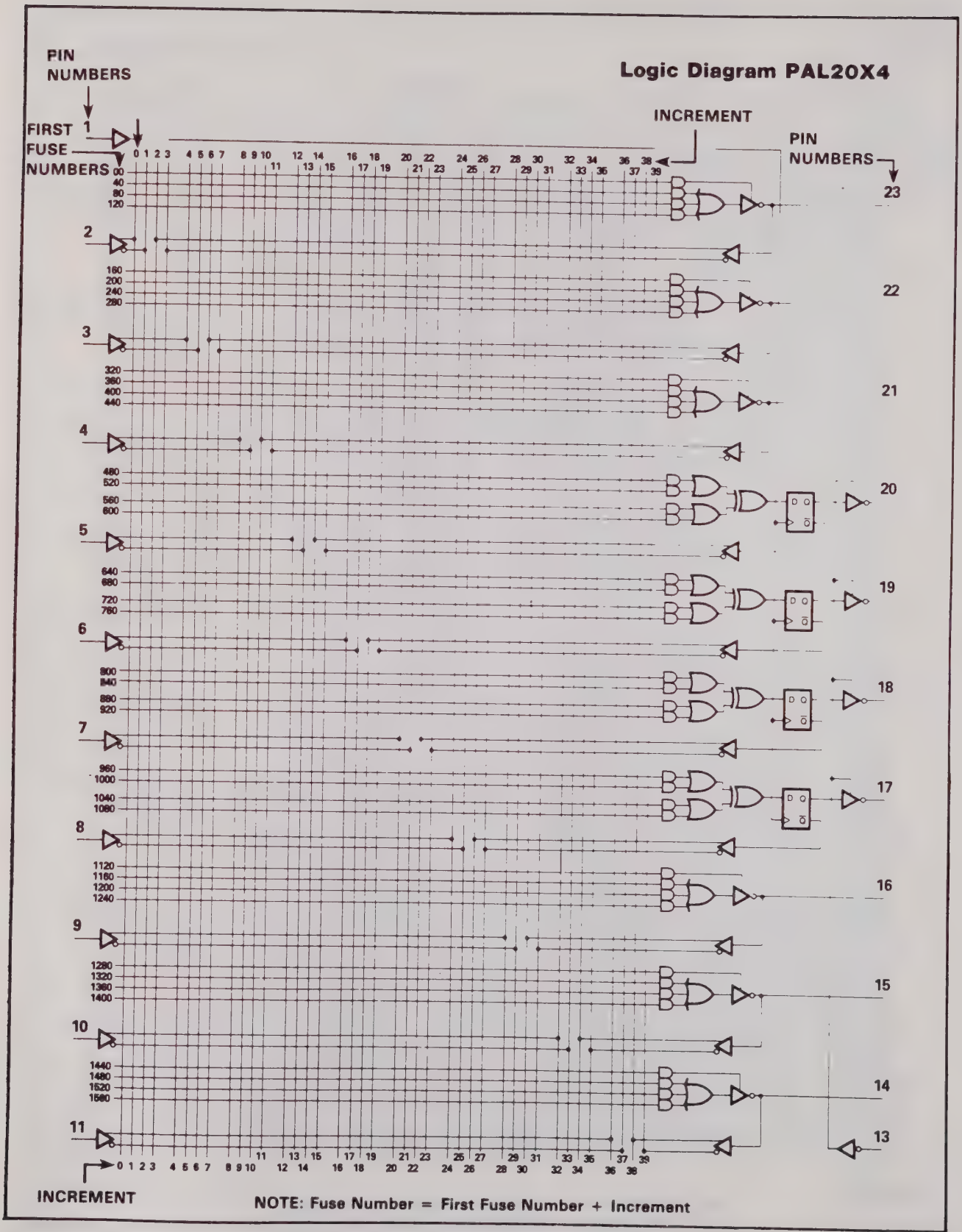


Figure B-35. Logic Diagram PAL20X4

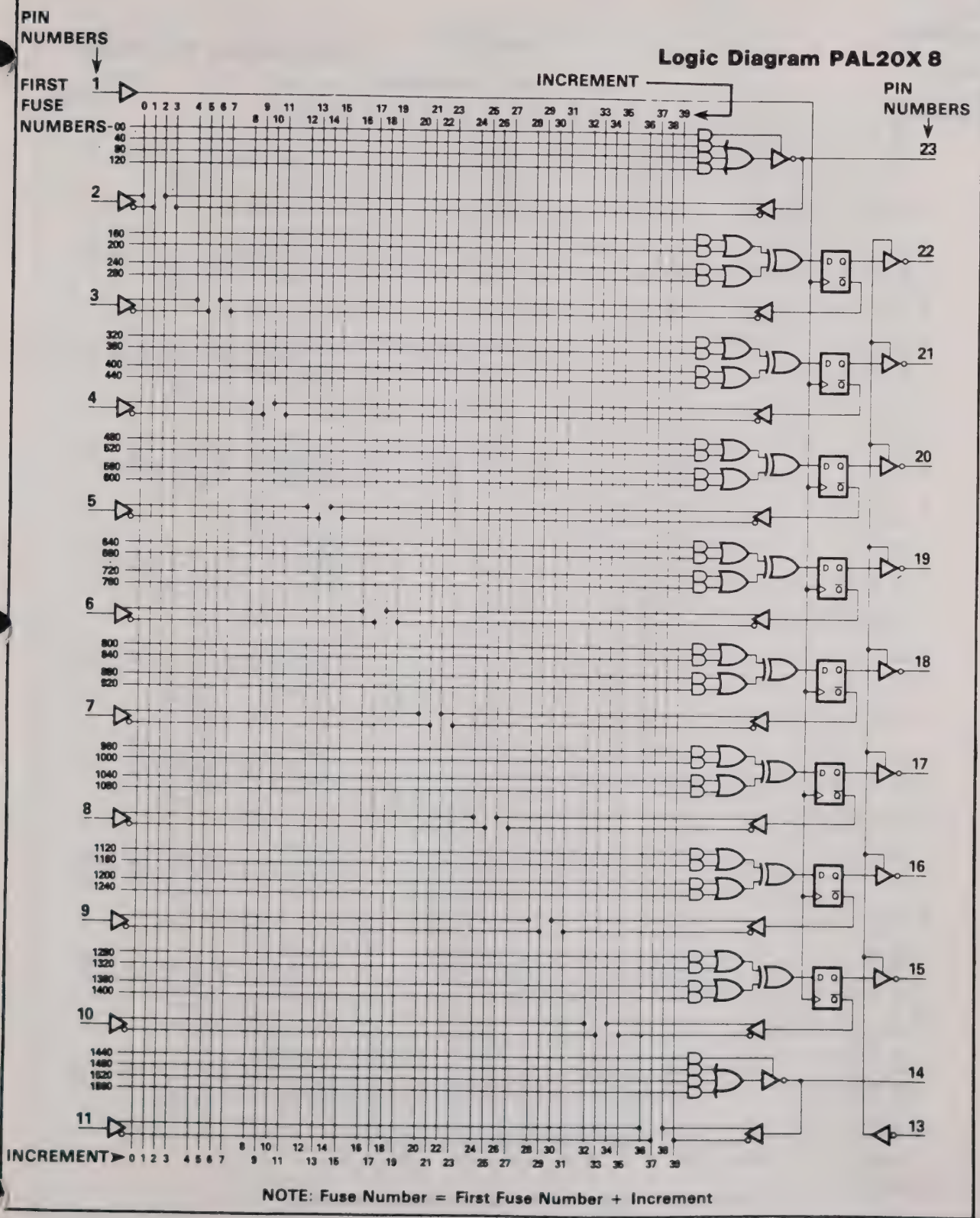


Figure B-36. Logic Diagram PAL20X8

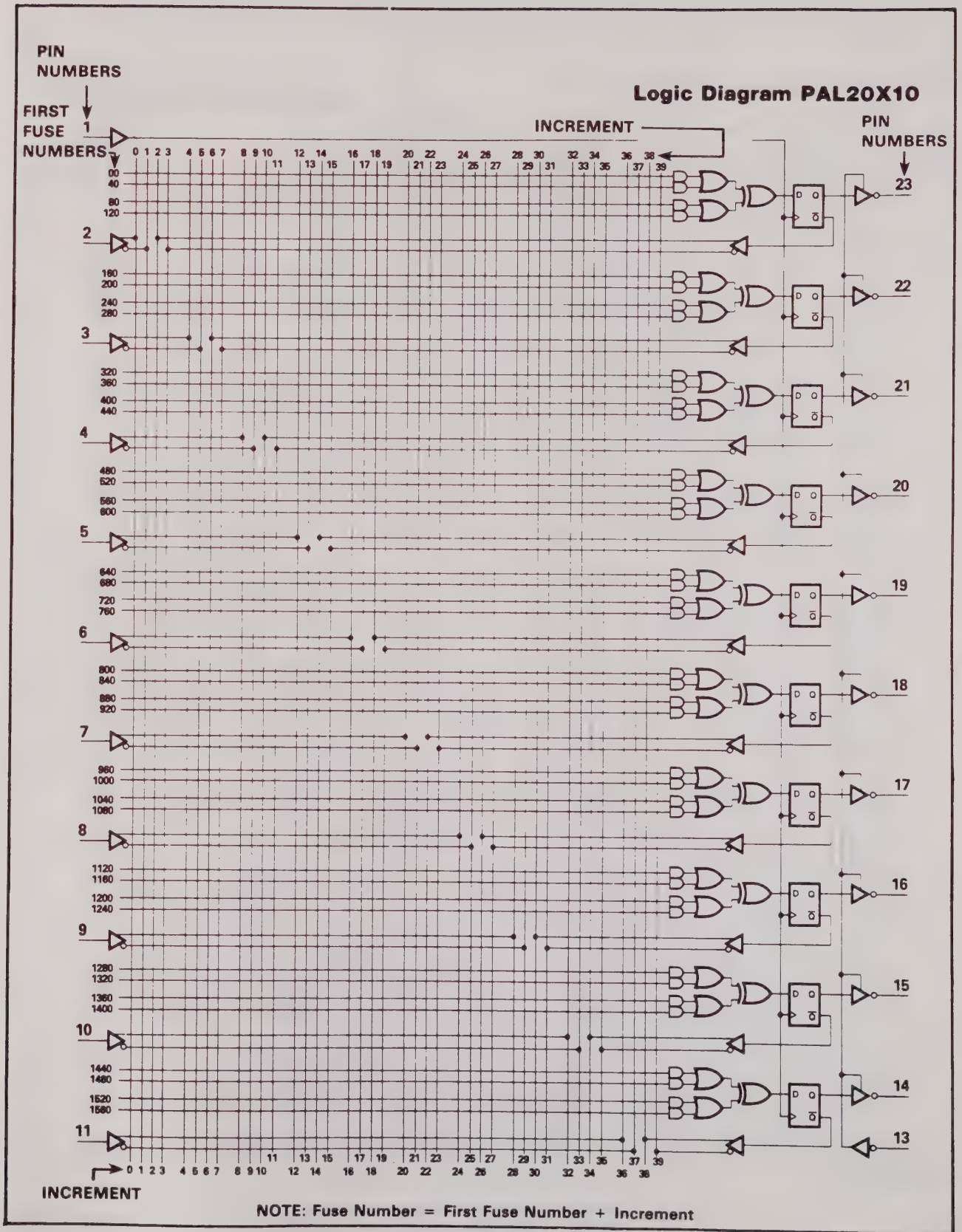


Figure B-37. Logic Diagram PAL20X10

SCHEMATIC

30-702-1951 Memory Board

REVISIONS					
ZONE	LTR	DESCRIPTION	CK	CM	PE	DATE
	A	RELEASE				7/16/82
	B	ECN # 4683				9/82
	C	ECN # 4684				9/82
	D	ECN 4787				8/83

P4

+5 — 1, 2, 3, 28 — C1-C5, C7, C8 .01; C6 22/15V

GND — 22, 23, 26, 71, 72

70 — 19 INTERLOCK

V·Φ2 — 25

A0 — 88
A1 — 82
A2 — 76
A3 — 67
A4 — 64
A5 — 85
A6 — 79
A7 — 73
A8 — 66
A9 — 68
A10 — 61
A11 — 65
A12 — 58

DO1 — 52
DO2 — 49
DO3 — 46
DO4 — 43
DO5 — 40
DO6 — 37
DO7 — 34
DO8 — 31

DEV — 29
SEL A — 14
SEL B — 13
A15 — 59
C3 — 47

R2 100
R1 100

SOFTWARE PROMS

U4, U3, U2, U8, U1 — 2732/2764 — VPP PGM — A0–A12 — OE CE — +5

U4: 8000 9FEF
U3: 8000 9FEF
U2: 8000 9FEF
U8: 8000 9FEF
U1: 8000 7FFF

U5 74LS32

DATA BUFFER

U7 74LS244 — OUT IN — EN EN

SECONDARY DECODE

U6 74LS138 — A, B, C — G1, G2A, G2B — Y0–Y3

U5 74LS32

C9 500 PF
C10 500 PF

NOTES: UNLESS OTHERWISE SPECIFIED.

1. ALL CAPACITORS ARE IN MICROFARADS, .1, 50V.
2. LAST REFERENCE DESIGNATION USED: C10, U8, P4, R2.
3. POWER & GROUND.

REF DES	+5	GND
U1-4, 8	26,28	14
U6	16	8
U7	20	10
U5	14	7

4. ALL RESISTORS ARE IN OHMS, ¼ W, 5%.

APPROVALS:	UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES	DATA I/O
PROJ. ENG. 7/16/82	TOLERANCES DECIMALS ANGULAR: X ±, XX ±, .XXX ±, DO NOT SCALE DRAWING	TITLE: SCHEMATIC DIAGRAM, DEVELOPMENT ADAPTER
ENG. MGR. 7/16/82		
Q.C. 7/16/82	DRAWN BY M Pattie 5/4/82	SIZE D; CODE IDENT NO 54193; DRAWING NO 30-702-1951
DATE MFG ENG 7/16/82	CHECKED BY Dan Cass 7/19	SCALE —; SHEET 1 1

U.S. and Canada Sales Offices and Representatives
Direct Sales and Service Offices

HAWAII, OREGON, WASHINGTON

Data I/O Corporation
10525 Willows Road N.E.
C-46
Redmond, WA 98052
(206) 881-6444
Telex 15-2167
(Sales & Service)

NORTHERN CALIFORNIA

Data I/O Corporation
473 Sapena Court
Suite 4
Santa Clara, CA 95050
(408) 727-0641
Telex 352054
(Sales, Applications & Service)

WESTERN REGION

Data I/O Corporation
2770 South Harbor Blvd.
Suite K
Santa Ana, CA 92704
(714) 662-1182
Telex 910-595-1562
(Sales & Service)

EASTERN REGION ADMINISTRATIVE OFFICES

MAINE, EASTERN MASSACHUSETTS, NEW HAMPSHIRE, VERMONT

Data I/O Corporation
Nashua Road
Route 101A
Amherst, NH 03031
(603) 889-8511 *(Sales)*
(603) 889-8513 *(Applications & Service)*
Telex 943431

SOUTHERN REGION ADMINISTRATIVE OFFICE

Data I/O Corporation
1810 N. Glenville Drive
Suite 108
Richardson, TX 75081
(214) 235-0044
Telex 792474
(Sales & Applications)

Representatives

(Sales Only)

ALASKA

Microprocessor Technology
3537 East 65th Avenue
Anchorage, AK 99507
(907) 344-8080
Telex (090) 25-462

OREGON, EASTERN WASHINGTON

Northwest Test and Measurement
8196 Southwest Hall
Suite 217
Beaverton, OR 97005
(503) 646-9966

ARKANSAS, LOUISIANA, OKLAHOMA, TEXAS

TESTECH, Inc.
1000 E. Campbell Road
Suite 108
Richardson, TX 75081
(214) 644-5010

9219 Katy Freeway
Suite 124
Houston, TX 77024
(713) 467-2192

2525 Wallingwood
Suite 804
Austin, TX 78746
(512) 327-7033

MICHIGAN, EASTERN KENTUCKY, OHIO, WEST VIRGINIA, WESTERN PENNSYLVANIA

Electro Sales Associates
1635 Mardon Drive
Dayton, OH 45432
(513) 426-5551

29200 Vassar Road
Suite 505
Livonia, MI 48152
(313) 474-7320

9816 Portage Road
Portage, MI 49002
(616) 323-2416

851 East 222nd Street
Cleveland, OH 44123
(216) 261-5440

3740 Mount Royal Blvd.
Allison Park, PA 15101
(412) 487-3801

UPSTATE NEW YORK

Martin P. Andrews
523 East Genesee Street
P.O. Box 443
Fayetteville, NY 13066
(315) 637-5291

METROPOLITAN NEW YORK, LONG ISLAND, NORTHERN NEW JERSEY

RTI
300 Northern Blvd.
Great Neck, NY 11021
(516) 829-3770

KANSAS, MISSOURI, NEBRASKA, SOUTHERN ILLINOIS

Midtec Associates, Inc.
7702 Mize Road
DeSoto, KS 66018
(913) 441-6565
Telex 910-743-4129

8460 Lindbergh North
Suite 11
Florissant, MO 63031
(314) 837-5200

ALABAMA, FLORIDA, GEORGIA, MISSISSIPPI, NORTH CAROLINA, SOUTH CAROLINA, TENNESSEE

Pen Tech Associates
1398 Semoran Blvd.
Suite 106
Casselberry, FL 32707
(305) 645-3444

201 S.E. 15th Terrace
Suite K
Deerfield Beach, FL 33441
(305) 421-4989

Cherokee Center
627 Cherokee Street
Suite 21
Marietta, GA 30060
(404) 424-1931

Holiday Office Center
3322 Memorial Pkwy. S.W.
Suite 36
Huntsville, AL 35801
(205) 881-9298

3709 Alliance Drive
Suite C
Greensboro, NC 27407
(919) 852-6000

INDIANA, IOWA, WESTERN KENTUCKY, MINNESOTA, NORTHERN ILLINOIS, NORTH DAKOTA, SOUTH DAKOTA, WISCONSIN

Torkelson Associates
4940 Viking Drive
Minneapolis, MN 55435
(612) 835-2414

108 Wilmot Road
Suite 110
Dearfield, IL 60015
(312) 945-8700

2346 S. Lynhurst
Room B101
Indianapolis, IN 46241
(317) 244-7867

2840 N. Brookfield Road
Brookfield, WI 53005
(414) 784-7736

ARIZONA, COLORADO, IDAHO, MONTANA, NEW MEXICO, NEVADA, UTAH, WYOMING

Zeus Electronics, Inc.
1428 E. Pierson Street
Phoenix, AZ 85014
(602) 263-6022
Telex 910-951-1362

310 Val Verde S.E.
Albuquerque, NM 87108
(505) 842-6633

3333 Quebec Street
Suite 2950
Denver, CO 80207
(303) 321-4246

1849 W. North Temple
Suite B105
Salt Lake City, UT 84116
(801) 534-0500

SOUTHERN NEW JERSEY, DELAWARE, MARYLAND, EASTERN PENNSYLVANIA, VIRGINIA, WASHINGTON D.C.

SCI-REP, Inc.
304 Cooper Center
Pennsauken, NJ 08109
(609) 662-5222

9512A Lee Highway
Fairfax, VA 22031
(703) 385-0600

CONNECTICUT, RHODE ISLAND, WESTERN MASSACHUSETTS

Pat Jenks Associates
52 Washington Avenue
North Haven, CT 06473
(203) 239-6201

114 Cummings Park
Woburn, MA 01801
(617) 938-0488

CANADA

Allan Crawford Associates, Ltd.
6503 Northam Drive
Mississauga, Ontario L4V 1J2
(416) 678-1500
Telex 06 968769

881 Lady Ellen Place
Ottawa, Ontario K1Z 5L3
(613) 722-7682
Telex 053 3600

7018 Cote De Liesse
St. Laurent, P.Q. H4T 1E7
(514) 731-8564
Telex 05-824944

1935-30th Avenue N.E.
Calgary, Alberta T2E 6Z5
(403) 230-1341
Telex 03 821186

3795 William Street
Burnaby, B.C. V5C 3H3
(604) 294-1326
Telex 04 54247

192 Joseph Zatzman Drive
Dartmouth, Nova Scotia B3B 1N4
(902) 463-9360
Telex 019 31604

15043A 118th Avenue
Edmonton, Alberta T5V 1H9
(403) 451-4893

Addresses on this list subject to change without notice.

International Sales Offices and Representatives

Europe

Data I/O Europe
Vondelstraat 50-52
NL-1054 GE Amsterdam
The Netherlands
(20)186855
Telex 16616 DATIO NL

Data I/O Germany GmbH
Bahnhofstrasse 3
D-6453 Seligenstadt
West Germany
(6182)3088/89
Telex 4184962 DATA D

AUSTRIA

Ing. Ernst Steiner
Hummelgasse 14
A-1130 Wien
Austria
(222)827474
Telex 135026 ES A

BELGIUM

Simac Electronics
Rue du Progres 52
BOITE 3
B-1000 Brussels
Belgium
(2)2192451-3
Telex 23662 SIMEIP B

DENMARK

ITT Komponent A/S
Fabriksparken 31
DK-2600 Glostrup
Denmark
(2)451822
Telex 33355 itt dk

FINLAND

Havulinna-Instrumentarium Oy
P.O. Box 357
SF-00101 Helsinki 10
Finland
(0)799711
Telex 124426 havul sf

FRANCE

M.B. Electronique
606, rue Fourny
Zac de Buc
P.O. Box 31
F-78530, Buc, France
(3)9568131
Telex MB 695414 F

GERMANY

Instrumatic Electronics GmbH
Am Kirchenholzel 14
D-8032 Grafelfing (near Munich)
West Germany
(89)852063
Telex 524298 INSTR D

GREECE

Mr. Elias Mammeas
65 Meg. Alexandrou & Athinas Street
P.O. Box 181
Korydallows
Piraeus
(1)4967815/or 818
Telex 213835 LH GR

ISRAEL

R.D.T. Electronics Engineering, Ltd.
46 Sokolov Street
P.O. Box 75
Ramat Hasharon
Israel
(3)483211
Telex 32143 RDT IL

ITALY

Sistrel SPA
Via Giuseppe Armellini 39
I-00143 Roma
Italy
(6)5915551
Telex 680356 SISTREL i

Sistrel SPA
Via P. da Volpedo 59
I-20092 Cinisello Balsamo (Mi)
Italy
(2)6120129 or 6181893
Telex 334643 SISTREL i

NETHERLANDS

Simac Electronics
Veenstraat 20
NL-5503HR Veldhoven
The Netherlands
(40)533725
Telex 51037 SIMAC NL

NORWAY

Teleinstrument A/S
P.O. Box 134
N-1371 Asker
Norway
(2)789460
Telex 72919 TELIN N

PORTUGAL

Decada
Equipamentos de Electronica, Lda.
Rua Pedro Nunes 47-C
P-1000 Lisboa or P-1003 Lisboa Codex
Portugal
(19)574984
Telex 18469 NONIO P

SPAIN

Instrumatic
Alameda Principal 26
Apartado 151
Malaga 5
Spain
(52)213199/213898
Telex 77131 Hafn e

Instrumatic
Juan Hurtado de Mendoza 9
Madrid-16
Spain
(1)2502577/2507278
Telex 46277 Itic e

SWEDEN

Macrotek AB
Vallingbyvagen 212, Box 43
S-162 11 Vallingbyvagen, Sweden
(8)870190
Telex 12543 MATEK S

SWITZERLAND

Instrumatic SA
5-7, rue du Clos
CH-1207 Geneve
Switzerland
(22)360830
Telex 28667 FLUJA CH

Instrumatic AG
Weingartenstrasse 9
CH-8803 Ruschlikon
Switzerland
(1)7241410
Telex 56605 INST CH

TURKEY

Data I/O Europe
Vondelstraat 50-52
NL-1054 GE Amsterdam
The Netherlands
(20)186855
Telex 16616 DATIO NL

UNITED KINGDOM

Microsystem Services
P.O. Box 37
Lincoln Road
Cressex Industrial Estate
High Wycombe
Bucks, HP12 3XJ, England
(494)41661
Telex 837187 MICSYS G

International

Data I/O International
10525 Willows Road N.E.
C-46
Redmond, WA 98052
U.S.A.
(206) 881-6444
Telex 15-2167

AUSTRALIA

CORPORATE OFFICE

Warburton O'Donnell Ltd.
372 Eastern Valley Road
P.O. Box 182
Chatswood, N.S.W., Australia 2067
4073261
Telex Warfran AA21299

SOUTH AUSTRALIA

ADELAIDE, AUSTRALIA

Warburton Franki (Adelaide) Pty. Ltd.
322 Grange Road
Kidman Park
South Australia 5025 Australia
3567333
Telex Warfran AA82579

BRISBANE, QUEENSLAND

Warburton Franki (Brisbane) Pty. Ltd.
13 Chester Street
Fortitude Valley
Queensland 4006 Australia
527255
Telex Warfran AA41052

MELBOURNE, VICTORIA

Warburton Franki (Melbourne) Pty. Ltd.
220 Park Street
South Melbourne
Victoria 3205 Australia
699 4999
Telex Warfran AA31370

PERTH, WESTERN AUSTRALIA

Warburton Franki (Perth)
98-102 Belgravia Street
Belmont, 6104
Western Australia
65 7000
Telex Warfran AA92908

SYDNEY, NEW SOUTH WALES

Warburton Franki Sydney
199 Parramatta Road
Auburn, N.S.W. 2144
Australia
648 1711
Telex Warfran AA22265

TASMANIA

Associated Agencies Pty., Ltd.
P.O. Box 252
43 Albert Road
Moonah, Tasmania 7009
23 1841
Telex ASSAG AA58206

BRAZIL

Cosele
Instrumentos Electronicos Ltda.
Rue da Consolacao, 867
3.° And. - Conj. 32
01301 - Sao Paulo - SP
Brasil
255-1733 or 256-7421 or 231-5548
Telex (011) 30869 CSEL-BR

HONG KONG

Cummings Engineering Co., Ltd.
90 Sung Wong Toi Road, I/F,
To Kwa Wan, Kowloon,
Hong Kong
3-636227
Telex 74015 CUMMH HX

INDIA

Transmarketing Private, Ltd.
Sterling Center
16/2 Dr. Annie Besant Rd.
Bombay 400-18
India
022 391874
Telex 011-5424

JAPAN

Data I/O Japan Company, Ltd.
Sankei Building
1-8, Sarugaku-cho 2-Chome
Chiyoda-Ku, Tokyo 101
Japan
(03) 295-2656
Telex 2225391 DATAIO J

KOREA

Elcom System, Inc.
Yeongdong Box 50
Seoul, Korea
555-5222 or 557-3836
Telex ADUCEL K25227

MALAYSIA

Cadco Technical Services
67A, Jalan Perisai
Taman Sri Tebrau
Johore Bahru, Malaysia
Cable - CADCOKIT - JOHOR BAHRU

MEXICO

Christensen, S.A.
Guillermo Prieto 76-304
Col. San Rafael
Delegacion Cuahutemoc
06470-Mexico, D.F.
546-25-95
546-29-55
Telex 017-75612 Mycome.

NEW ZEALAND

Wellington-New Zealand:
Warburton Franki, Ltd.
42-43 Oxford Terrace
Lower Hutt, New Zealand
693-016
Telex Warfran NZ 3824

Warburton Franki, Ltd.
P.O. Box 9301; Newmarket
142 Broadway
Auckland, New Zealand
504-458
Telex Warfran NZ 3824

PHILIPPINES

Cummings Engineering Co., Ltd.
90 Sung Wong Toi Road, 1/F,
To Kwa Wan, Kowloon
Hong Kong
3-636227
Telex 74015 CUMMH HX

SINGAPORE

GEA Technology PTE., Ltd.
Units 1003 to 1008, Block 3, 10th Floor
PSA Multi-Storey Complex
Pasir Panjang Road
Singapore 0511
2729412
Telex RS 37162 Answerback Geasin

SOUTH AFRICA

Electronic Building Elements
(PTY) Ltd.
P.O. Box 4609
Pine Square, 18th Street
Hazelwood
Pretoria, South Africa 0001
46-9221/7
Telex 3 0181 SA
Telegrams Elbilem

SOUTH & CENTRAL AMERICA

Dexter Computer Products, Inc.
3465 Torrance Blvd.
Suite N
Torrence, CA 90503
213-542-7333
Telex 182360 DXTR TRNC

TAIWAN

COMPAC Microelectronics, Inc.
11th Fl., Walsin Bldg.
#219, Chung-Hsiao E. Road, Sec. 4
P.O. Box 53-479
Taipei, Taiwan R.O.C.
7529911
Telex 24368 COMTWN

THAILAND

Dynamic Supply Engineering R.O.P.
12 Soi Pasara 1, Sukhumvit 63
Bangkok-11, Thailand
Tel. - 3914434, 39228532
Telex 82455 DYNASUP TH

ATTENTION
MANUAL CHANGE INFORMATION
INSTRUCTIONS

At Data I/O, we continually strive to improve our equipment by adding new improvements to our hardware and software as soon as they are developed and tested.

The attached Documentation Change information reflects an update to this equipment. Please file the Document Change Summary and Document Change Revised Pages Record forms in front of the Title Page in your manual. The format of the Document Change information you receive will vary. Follow the instructions below for the correct procedure.

1. Document Addendum: File Document Addendums behind the Document Change Revised Pages Record.
2. Revised Pages: Replace the existing pages in the manual with the revised pages.
3. Complete Revision: Remove all existing pages, including Document Addendums, and replace them with the newly issued revision.

If you have any questions regarding these changes, please contact:

Ms. Jan Simmons
Technical Publications Supervisor
Data I/O Corporation
10525 Willows Road NE/C-46
Redmond, WA 98052
(206) 881-6444
Telex 15-2167

DOCUMENT CHANGE SUMMARY

At Data I/O, we continually strive to improve our instruments as soon as they are developed. Sometimes, because of printing and shipping requirements, we can't include these changes immediately into printed manuals. Consequently, your manual may contain new change information as summarized below.

File the Documentation Change Summary and the Document Change Record pages in the beginning of your manual, in front of the Title Page. File any Document Addendum pages received with changes immediately after the other two Document Change pages. If you have received revised pages, remove the previous page version, and replace it with the latest page revision. If you have received a complete revision, remove all previously filed Document Addendums and replace the existing manual pages with the revised manual pages. Whenever you receive a manual update, replace the existing Document Change pages with the most recent set of Document Change pages.

REV	DESCRIPTION OF CHANGE	DATE
B	Incorporate ECN# 4961	10/83

0647/1282

DOCUMENT CHANGE REVISED PAGES RECORD

REVISED				ADDED		DELETED	
PAGE NUMBER	REV & DATE	PAGE NUMBER	REV & DATE	PAGE NUMBER	REV & DATE	PAGE NUMBER	REV & DATE
Title Page	B 10/83						

0546/1282

K&M Computing
TORRANCE, CALIFORNIA 90505

D311-10L